Image Collection Optimization in the Design and Operation of Lightweight, Low Areal-Density Space Telescopes

by

Josef Roach Bogosian

Submitted to the Department of Aeronautics and Astronautics
in partial fulfillment of the requirements for the degree of

Master of Science in Aeronautics and Astronautics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY
[June 2008]
May 2008

Author .....
Department of Aeronautics and Astronautics
May 23, 2008

Certified by .....
David W. Miller
Professor
Thesis Supervisor

Accepted by .....
Prof. David L. Darmofal
Associate Department Head
Chair, Committee on Graduate Students

Image Collection Optimization in the Design and Operation of Lightweight, Low Areal-Density Space Telescopes

by

Josef Roach Bogosian

Submitted to the Department of Aeronautics and Astronautics
on May 23, 2008, in partial fulfillment of the
requirements for the degree of
Master of Science in Aeronautics and Astronautics

Abstract

Demand for space imagery has increased dramatically over the past several decades. Scientific and government agencies rely on Earth-observing space assets for a variety of functions, including mapping, agriculture, and intelligence. In recent years, online interactive mapping services have created a large demand for high-resolution commercial satellite imagery. The satellite systems launched to meet the demand for imagery have two major objectives: 1) efficient global Earth coverage and 2) responsiveness to real-time events. Depending on the specific application, mission architects may particularly value one objective. Commercial satellites need to fulfill tasking requests from customers and are primarily focused on global accessibility and efficient imaging. Engineers may design military or environmental warning satellites, on the other hand, to focus on quickly responding to events in unpredictable locations.

This thesis investigates two elements in support of the design of Earth observing satellite systems. The first part is a study of a responsive satellite constellation architecture. The focus within the Responsive Space community has primarily been on small, lightweight, disposable satellite systems. Industry and academia have done less work to consider architectures that meet the responsiveness objective while still providing global coverage with sustainable orbits. This thesis analyzes an architecture that supports objectives of efficient coverage of the globe and also responsiveness to arising targets.

The space community has also demonstrated significant interest in lightweight space telescopes. These systems offer launch cost savings and, in the case of segmented aperture optics, can be stowed and deployed on orbit. The reduction in mass comes, however, at the price of structural flexibility, which affects the satellite's ability to efficiently image targets. The second part of this thesis explores how satellite dynamic properties affect the ability to provide efficient imaging. Satellite scheduling optimization formulations, including graph search, integer programming, and dynamic programming, enable evaluation of imaging efficiency. Integration of imaging performance metrics into a trade-space analysis tool allows for more informed decisions early in the satellite design process.

Thesis Supervisor: David W. Miller
Title: Professor

Acknowledgments

The author would like to thank the Department of Defense for funding this research; Professor David W. Miller for his guidance; Thomas Gray and Lucy Cohan for their help on MOST; and Rebecca Myers and Benjamin Werner for their work on the RECON project. The author would also like to especially thank his family and friends for all of their support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

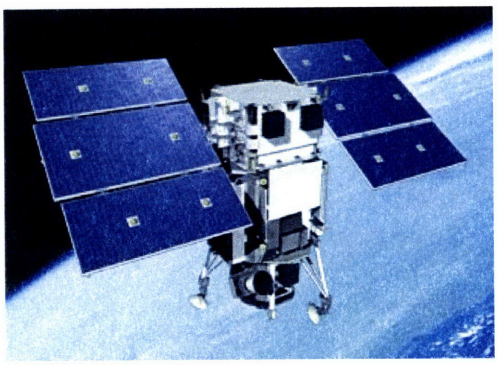
Reliance on space resources has increased dramatically over the past several decades. Scientific agencies rely on satellites for geographical and meteorological data. Government agencies likewise rely on satellites for accurate real-time intelligence. Satellite imagery finds application in areas such as agriculture, urban planning, disaster relief, and navigation. In recent years, interactive mapping services, such as Google Maps [25], have greatly increased the demand for high-resolution commercial satellite imagery. For all of these applications, there are two main design objectives for the systems which provide imagery.

1. **Efficient coverage of all areas of interest.** Satellite systems should have the capability to image a large area (or series of areas) in a short time period. This objective is important for scientific, defense, and commercial applications.

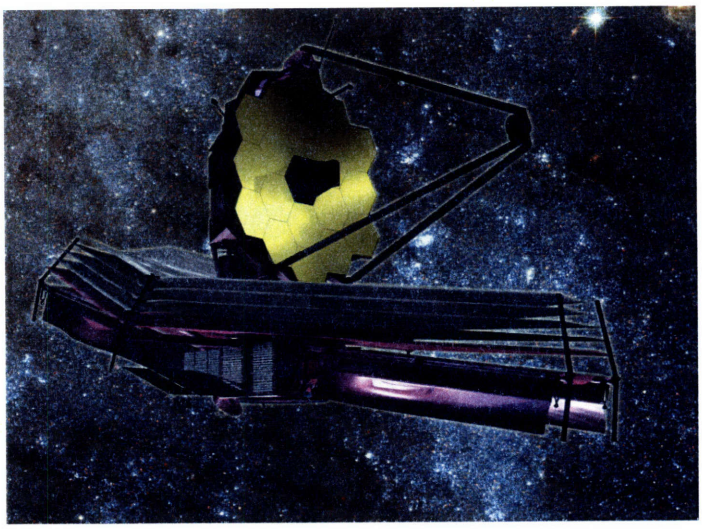
2. **Responsiveness to arising real-time events.** Satellite systems should collect images of an area of interest, whose location is not known ahead of time, as soon as possible following the event. Responsive imagery provides useful and actionable information. Imagery users also desire repeated coverage of the area following the initial observation. Environmental warning and defense organizations place particular value on this objective. The mission may be to survey a natural disaster, for example a hurricane or tsunami, or gather intelligence from a military event such as a missile launch.

Each imagery application requires the fielding of satellite systems that meet specific design objectives. In general, there is a large spectrum of needs for different satellite systems. Imagery demands for applications with minimal dynamics (agriculture, urban planning,

(a) IKONOS Imaging Satellite [30]



(b) WorldView-1 Imaging Satellite [69]



(c) James Webb Space Telescope (JWST) [32]

Figure 1-1: Modern Space Imaging Systems

geography) can be fulfilled by efficient but non-responsive satellites. More agile systems are needed to fulfill demands for imagery in defense and natural disaster scenarios.

In the science community, the NASA LandSat program has provided Earth imagery since the early 1970's [36]. In recent years, satellite systems such as SPOT-5, IKONOS, and WorldView-1 have provided commercial imagery [57, 31, 69]. Engineers designed these systems primarily to meet the first main objective, efficient Earth coverage, subject to cost constraints. While not explicitly 'responsive' satellites, the orbit design of each provides a low revisit time - on the order of days - to any target. Demand continues to grow for Earth imagery, and plans exist for continuing expansion of Earth-observing resources.

The aerospace industry has demonstrated interest in building systems that specifically address the second main objective. The field of *Responsive Space* has developed in the past decade in an effort to create an agenda and requirements for missions of this type, and encourage industry to pursue responsiveness [52, 45]. Researchers have proposed several mission architectures, including a 'Launch-on-Demand' approach. Another suggested option is to use existing on-orbit assets, reconfiguring or re-tasking the satellites as necessary to focus on the area of interest.

1.1 Motivation

This thesis investigates two elements in support of the design of Earth observing satellite systems. The first is a Responsive Space architecture study. The focus within Responsive Space has predominantly been on lightweight, small, disposable satellite systems built exclusively for the purpose of responsiveness. Researchers have done less work to consider architectures which meet the responsiveness objective while still providing global coverage with sustainable orbits. This thesis proposes a system architecture that supports objectives of efficient coverage of the globe and also responsiveness to arising targets.

The second part of this thesis focuses on the design of lightweight space telescopes in support of the efficient imaging objective. While satellites such as the James Webb Space Telescope (JWST) (Figure 1-1(c)) will demonstrate advanced lightweight technologies for astronomical applications, the same performance benefits have great potential to translate to lightweight Earth-observing systems. Interest in lightweight space systems has prompted the development of a trade space exploration tool, MOST [60, 61, 15], at the MIT Space Systems Laboratory [47]. The slew and settle performance metric generated by the MOST model serves as a useful approximation of the satellite operational performance, but it is important to consider how the slew and settle metric translates to imaging efficiency. Problem formulations for the optimization of Earth imaging enable analysis of satellites with different agility properties. Incorporation of this analysis into a satellite trade-space analysis tool allows for more informed design decisions.

1.2 Approach

This thesis investigates two aspects in the design of efficient and responsive Earth observing systems. The first part introduces a system architecture that meets the objectives of Responsive Space while also providing comprehensive coverage of the globe. The second part discusses an approach for parametrically evaluating the imaging performance of lightweight Earth observing systems. The following sections provide an outline of this thesis.

1.2.1 Reconfigurable Satellite Constellations

Recently there has been a push in academia and industry for the investment in, and development of, responsive space systems. Responsiveness is, in general, the ability to respond to a rapidly changing and unpredictable target. Several architectures have emerged as having potential to meet the objectives of Responsive Space [67, 59].

One architecture features on-orbit assets which are tasked as needed to respond to events. This responsiveness may be built in to the original constellation orbit design, or there may be explicit procedures, such as orbit reconfiguration, that occur in response to target notification. A second 'Launch-on-Demand' architecture involves building a fleet of satellites that are placed in storage on the ground. Upon notification of a target, the satellites can be launched in 12-24 hours. One or more launches occur to place the satellites in orbits that provide good coverage of the target of interest. Mission lifetimes for the second architecture are often very short, on the order of weeks to months. A third architecture is 'Build-on-Demand.' The operating method here is to develop procedures which enable the rapid manufacture (and possibly modification) of satellite systems. The responsiveness of this architecture is considerably slower, but the advantage is that the risk of having completed satellites remain on the ground is eliminated. Finally, aerial reconnaissance is a responsive architecture with very different performance properties. Depending on the sensitivity of the mission, the vehicles may be manned or unmanned, and may or may not have stealth capabilities.

This thesis investigates an architecture of the first type, a Reconfigurable Constellation architecture abbreviated RECON. The RECON architecture meets the objectives of long-term global imaging, while also having the capacity to respond to real-time events. A series of satellites is launched into a nominal Earth observing constellation, which provides global

Earth coverage. Upon notification of a specific target, mission operators reconfigure the constellation into a regional observing mode. The regional observing mode offers increased coverage and lower revisit time for the target of interest. After the designated mission duration, the satellite operators return the constellation to the nominal global observing mode.

A cost model that characterizes the major elements the RECON architecture facilitates analysis of different designs. The cost model has the capability to generate a set of different constellations of equal cost. Performance comparisons between constellations within this set yield information about the value (performance per cost) of each constellation. Tools developed in MATLAB® [2] and Satellite Tool Kit™[3] enable accurate evaluation of each mode of the satellite constellation.

Extensions are pursued to allow more rapid analysis of RECON architectures by eliminating the need for time simulation of the constellations. Performance metrics for the global observing mode constellation enable approximate results in markedly shorter time. The full problem, including the cost model and constellation performance metrics, is a mixed-integer non-linear optimization problem. The objective function accounts for coverage performance in global and regional observing modes, as well as a measure of responsiveness. The solution of the optimization problem is the best performing constellation for a fixed cost. Finally, comparisons between optimal RECON systems and an alternative responsive Unmanned Aerial Vehicle (UAV) fleet demonstrate the strengths and weaknesses of each approach.

This part of the thesis is broken down as follows. Chapter 2 introduces the RECON architecture, discusses the cost model, and presents the performance results of a set of equal-cost constellations. Chapter 3 presents an approach that reduces the analysis time for the RECON architectures using approximate coverage metrics. Chapter 3 also discusses the feasibility of a responsive UAV fleet as a RECON alternative.

1.2.2 Imaging Productivity of Lightweight Space Telescopes

The second part of this thesis extends the work in the first part by focusing on the design of an individual lightweight space telescope as it relates to imaging efficiency. The aerospace community has demonstrated interest in space telescopes that feature advanced lightweight mirror technology. These satellites offer savings in launch costs and, in the case of segmented aperture systems, can be stowed to reduce launch volume and deployed on orbit. By virtue

of being very lightweight, these systems are also significantly more structurally flexible than traditional massive monolithic designs. While this property can be modified by sophisticated control systems, the problem of flexibility is difficult to eliminate.

Satellite structural flexibility is important in determining the agility of the satellite. Slewing maneuvers excite vibration modes of the satellite that must settle before sufficient accuracy is available to take an image. Very flexible systems require a longer settle time, reducing the total available imaging time. For astronomical telescopes, exposure times for images are very long - on the order of days to weeks - and thus the fraction of time spent slewing the satellite is minimal. For Earth observing systems, which image many geographically distributed targets, the time spent slewing may be a large fraction of the total time. The total time required (slew and settle) to reorient the satellite has a large impact on the efficiency of the system.

The imaging optimization problem relevant for Earth observing systems allows estimation of this impact. The objective of the optimization problem is to find, for a given imaging window, the highest valued sequence of targets to image subject to dynamic constraints on the satellite. A variety of tools, including graph search, linear programming, and dynamic programming, can provide solutions to the problem. The optimized performance for different satellite systems, each characterized as a function of agility, illustrates the comparative value of the systems. The objective of the optimization problem is to find, for a given imaging window, the highest valued sequence of targets to image subject to dynamic constraints on the satellite.

Advanced optical systems often feature fast steering mirrors that help eliminate wavefront error and improve image quality. These steering mirrors have a limited capacity to image targets which are slightly off of the telescope boresight - allowing extension of the field of regard of the optics without requiring a slewing maneuver for the entire satellite. The range is considerably larger for systems, such as the SPOT [57] series of satellites, that utilize a steering mirror in front of the primary mirror. While steering mirrors increase the complexity of the optical system, they also increase the imaging productivity. The imaging optimization problem formulation must be modified to account for the performance benefits of the steering mirror. Dynamic programming and mixed-integer linear programming (MILP) tools provide optimal solutions, though each has some limitations.

Finally, this thesis discusses integration of the analysis procedure into the Modular

Optical Space Telescope (MOST) model, being developed at the Massachusetts Institute of Technology Space Systems Laboratory. MOST is a trade-space exploration tool for lightweight space telescopes. Parameters such as the mirror type (monolithic versus segmented) and f-number are adjusted within the model. For each design, an FEM is created and analyzed using NASTRAN to determine the vibration properties. The model estimates performance metrics such as slew and settle time and control effort to find optimal designs. The slew and settle time metric is an important measure of system performance, and the image collection optimization problem is applied to transform this information into more accurate operational imaging productivity metrics.

The lightweight telescope imaging productivity portion of the thesis is organized as follows. Chapter 4 introduces the Image Collection Optimization problem. Sections 4.1 through 4.3 provide an introduction to the problem and a review of existing literature. Section 4.4 discusses three methods for finding optimal solutions and compares the performance of each. Section 4.5 discusses extensions for the steering mirror problem. Chapter 5 explains the integration of this analysis into the MOST model. Section 5.1 describes the application of the tools developed in Chapter 4 to compare different satellite designs. Section 5.2 compares several satellite designs for the purpose of demonstrating the influence of satellite agility on imaging productivity. Finally, Section 5.3 describes integration with the MOST model and makes recommendations for improving analysis efficiency.

Chapter 2

Reconfigurable Constellations (ReCon)

Modern defense and environmental agencies desire surveillance of specific targets and areas of interest. Ideally airborne or spaceborne assets perform this surveillance in real-time. In the case of space systems, large satellite constellation costs inhibit the potential for persistent coverage. More frequently, a small set of space assets gradually acquires imagery. Engineers have traditionally designed satellite constellations to allow for steady, predictable coverage of certain regions of interest. Recently industry and academia have demonstrated interest in developing systems that can respond to emergency events, providing excellent coverage on short notice. This movement has grown under the title of *Responsive Space.* Static satellite constellations can provide support for long term collection objectives, but may not be very responsive. The objective of responsive systems is to be able to quickly respond to events around the world. Given this narrow objective, responsive systems may not be able to provide efficient global imaging.

This section focuses on a hybrid Reconfigurable Constellation (RECON) architecture, which meets the goals of both global coverage and responsiveness. A nominal Earth observing satellite constellation provides global imaging. In an emergency, mission operators reconfigure the constellation using in-plane altitude change maneuvers, to provide increased access to the target of interest. This regionally focused mode utilizes repeating ground tracks for frequent target revisits with high resolution imaging opportunities. The objective of this chapter is to introduce the RECON architecture and present an approach for modeling and

evaluating different systems within the architecture.

2.1 Introduction

Responsive Space has garnered significant attention in the last several years [52]. Industry has identified a need for developing systems which have a capability to perform well in scenarios where targets are mobile or unpredictable. Researches have suggested many different mission architectures, and there is an ongoing effort to develop enabling technologies and processes [11, 67].

There are three major approaches to Responsive Space [53]. The first is a constellation reconfiguration approach. A satellite constellation **A** provides a comprehensive Earth observation capability. When an emergency event occurs, mission operators maneuver the satellites into constellation **B**, which is better suited for observation of the target.

Launch-on-Demand is a second responsive architecture [59]. A series of satellites remain in storage on the ground. When necessary, the participating organization makes preparations and launches the satellites into orbits which provide good coverage of the target. Small, low-cost satellites are a common choice for this architecture due to the unpredictable nature of the launch scenarios. Orbital altitudes may be very low to improve imaging resolution, with mission lifetimes only on the order of weeks to months.

Finally, researchers have considered a concept where satellites are rapidly built in eight to twelve months following initial notification of mission requirements. While this approach sacrifices the immediate availability of imagery, the satellites may be tailored to meet specific goals. Additionally, this concept does not require premature financial investment into a system that may remain unused, as is the risk for Launch-on-Demand satellites.

This chapter focuses on a Reconfigurable Constellation (RECON) architecture. This architecture supports two objectives: passive long-term global imaging of the Earth and responsive short-term focused coverage of specific areas of interest. A nominal satellite constellation provides global observation access and general Earth imaging. When a target arises, ground operators reconfigure some or all of the assets into a regionally focused constellation. The regional constellation utilizes repeating ground tracks to enable frequent coverage of the target of interest. After interest in the target has subsided, mission operators return the repositioned satellites to the original constellation to resume general Earth

imaging.

2.2 Literature Review

In order to evaluate constellation reconfiguration, it is important to select both a suitable constellation design and suitable reconfiguration procedure. This section outlines previous work performed on each of these fronts, and motivates the pursuit of the RECON architecture.

Researchers have given considerable attention to satellite constellation design over the last several decades [65]. The objective of these studies has been primarily to provide global or regional coverage with the minimum number of satellites. Walker proposed constellations to provide coverage within a given latitude band [64]. The Streets of coverage methodology utilizes polar orbits to provide global coverage [5]. Both configurations can provide *continuous* global coverage, or accumulated global coverage over days, weeks, or months. Draim et al. have proposed using Highly Elliptical Orbit (HEO) satellites to provide global or regional coverage [17]. Recent work has further addressed this concept [35]. Work has been done to optimize coverage using these individual architectures. Recent research has focused on responsive orbits and the advantages of responsive systems in providing surveillance [66, 37]. Wertz described research on agile spacecraft that have a large ΔV capability to make orbital manuevers [67].

Researchers have also studied optimal constellation reconfiguration [56]. In particular, Scialom focused on minimizing ΔV during the reconfiguration, while considering other metrics such as reconfiguration time, cost, and coverage levels. The research assumed that the initial and final constellation arrangements had been determined by separate analysis.

The work presented in this thesis combines elements of constellation design and reconfiguration to provide a unified Responsive Space architecture.

2.3 Methodology

This thesis explores many design parameters in determining the feasibility of the RECON architecture. A model for the global and regional coverage modes of the satellite provides performance information. The design attributes of each mode incorporates previous research in the area of constellation design. The reconfiguration procedure and ΔV require-

ments, determined based on relevant orbital dynamics, enable comparison of constellation responsiveness.

A cost model for a satellite constellation facilitates objective comparison of different designs within the RECON architecture. The cost model includes satellite development, launch, and operations costs, as well as the costs of constellation reconfiguration. Utilization of the model to generate sets of equal cost systems is a key to the analysis. Since the constellation cost has been fixed as a basis for distinguishing between system designs, a performance comparison of the resulting constellations enables isolation of specific design properties as the source of value. Systems which deliver the most value can be identified.

A MATLAB® and STK™ interface enables rapid simulation of satellite constellations [2, 3]. MATLAB® tools analyze regional observation mode performance metrics. Several main figures of merit are important when comparing the performance of different satellite constellations. Three primary figures of merit are described here:

- Access Time: The amount of time that a target is in view per time period. Also referred to as 'Coverage'.
- Revisit Time: The average length of time between consecutive accesses to a target.
- Response Time: The amount of time between notification of a target and the first image taken in regional observing mode. Generally quoted as maximum response time.

The access time and revisit time are relevant for both modes of the constellation. The response time is most relevant to the reconfiguration process. The integrated model is employed to analyze the architecture design, demonstrate the analysis process, and provide insight into the properties of the best performing constellations. In addition to simulation methods, Chapter 3 explores analytical estimates for each of these figures of merit to support a more rapid end-to-end optimization.

The analysis is presented as follows. Section 2.4 presents the RECON architecture in detail, including definitions of the global and regional observing modes, and an outline of the reconfiguration process. Section 2.5 presents a cost model that is used to approximate the cost of launching and operating a RECON satellite constellation as a function of several design parameters. Section 2.6 uses the cost model to identify a set of equal cost RECON

systems. Section 2.7 presents performance analysis of a subset of these equal cost satellite constellations, utilizing both analytical relationships and simulation tools. Finally, Section 2.8 provides conclusions regarding the value of the RECON architecture.

2.4 System Architecture

The RECON architecture features global and regional observing modes. This section introduces the properties of each mode and describes the process for transitioning between these modes.

A discussion of several important geometric parameters is important before describing details of the architecture. Figure 2-1 illustrates important Earth observing system geometry. For a satellite at an altitude h with an Earth-viewing off-nadir pointing angle θ, Equation (2.1) describes the minimum elevation angle ϵ, where R_E is the radius of the Earth.

$$cos(\epsilon) = sin(\theta)\frac{R_E + h}{R_E} \tag{2.1}$$

Equation (2.2) gives the Earth central angle λ.

$$\lambda = \frac{\pi}{2} - \theta - \epsilon \tag{2.2}$$

The satellite field of regard footprint is the region over which the satellite can slew and still observe the Earth's surface. Equation (2.3) calculates this value using relationships for a spherical cap.

$$A_{FOR} = 2\pi R_E^2(1 - cos(\lambda)) \tag{2.3}$$

Finally, the range to the target, h', is described by Equation (2.4).

$$h' = R_E\frac{sin(\lambda)}{sin(\theta)} \tag{2.4}$$

These relationships are fundamental to the development of the cost model and analytical relationships for coverage performance.

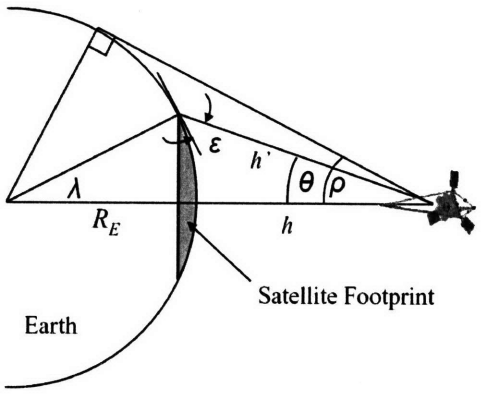


Figure 2-1: Earth Observation Geometry

2.4.1 Global Observing Mode Constellation

The Global Observing Mode (GOM) of the RECON architecture supports the objective of global coverage over time. The word 'global' is used here to mean 'over all possible target locations'. In this thesis, the possible target locations are between approximately $-60°$ and $60°$ latitude, which holds a significant portion of the Earth's landmass and population.

Section 2.2 introduced several design possibilities. Walker constellations and Streets of Coverage constellations are two traditional approaches. While the Streets of Coverage configuration is feasible, it apportions a significant amount of unneeded coverage to the area at the poles. Walker patterns are ideal for providing coverage within certain latitude bounds, and additionally can be established at a flexible range of altitudes.

Wertz reviews five different 'responsive' orbits [66]. Both HEO Cobra and Magic orbits are responsive, but require orbital altitudes which are out of the range necessary for high resolution observations. This is a result of the fact that ground resolution quickly degrades with increasing altitude for a fixed angular resolution requirement. Low Earth Repeat Coverage Orbits are specialized for known target locations, and require precise orbit inclinations. Low Earth Sun-Synchronous and Low Earth Fast Access Orbits, however, have the useful features of being feasible at promising design altitudes, and being flexible for incorporation into a full constellation.

The RECON global observing mode utilizes a Walker pattern constellation, an example of which is shown in Figure 2-2. The orbits are circular and in proximity to repeating ground track orbits to be discussed further in Section 2.4.2. A Walker constellation provides

coverage with a minimal amount of satellites, and is described by five variables [64]:

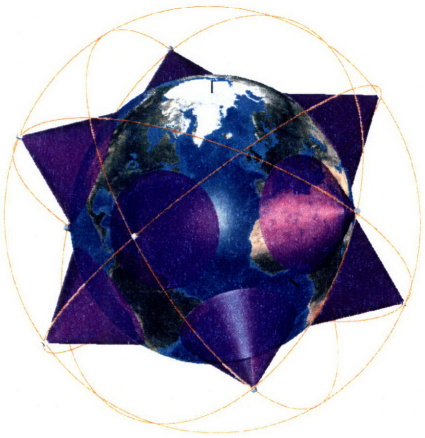


Figure 2-2: Sample Walker Constellation

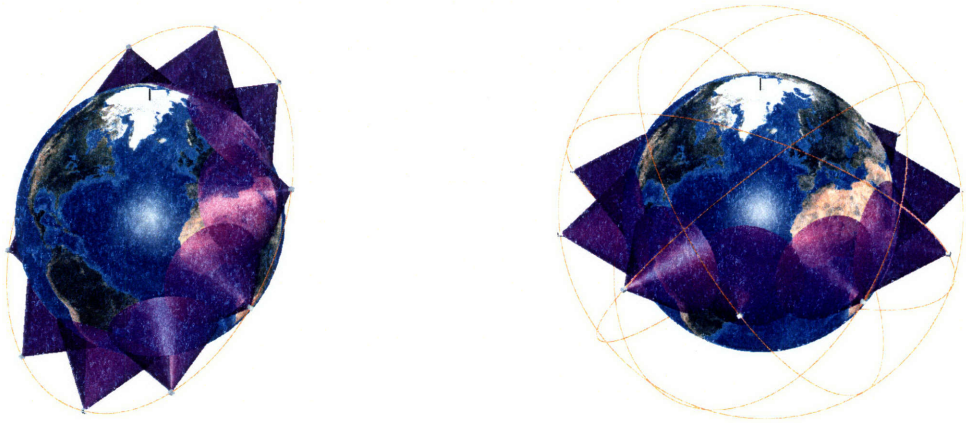
- i - Inclination of all constellation satellites
- h - Altitude of the constellation
- N - Total number of satellites
- P - Number of evenly spaced planes in which the satellites are placed
- f - The phase difference of each satellite with the nearest satellite in the adjacent plane

The inclination of the constellation determines the area of the Earth's surface which is accessible. An inclination i results in coverage of latitudes approximately between $-i$ and i. With an appropriate choice of altitude, the constellation sweeps out all area within the designed latitude bounds over time.

For the analysis in this thesis, the parameter P is fixed at $N/2$, that is the constellation is comprised of $N/2$ planes, each with two satellites. The purpose of this choice is to ensure that coverage of a given target while in regional observing mode is dispersed evenly throughout the day. All observation opportunities for a specific target are the result of the rotation of the Earth through a satellite orbital plane. A single plane of satellites can offer only two imaging opportunities per day. Increasing the number of satellites on a single plane above two does not offer any improvements in coverage distribution of the desired target. The only gains that can be made on a single plane are the result of clustering many

satellites to increase the target time in view near each imaging opportunity. The average revisit time for the one plane constellation remains nearly a half a day.

A Figure 2-3 illustrates two alternatives for placing $N = 10$ satellites in a Walker constellation. If all N satellites are placed in a single plane ($P = 1$), for example, there are only two observation opportunities per day - corresponding to the target rotating through the plane of satellites. The choice of $P = N/2$ distributes the N satellites maximally across the globe, reducing the average revisit time.



(a) Walker Constellation with $N = 10, P = 1$ [4]

(b) Walker Constellation with $N = 10, P = 5$ [4]

Figure 2-3: Alternate Walker Constellation Configurations for $N = 10$ Satellites

The performance of the Walker patterns described here is evaluated via simulation using Satellite Tool Kit (STK)^{TM} in Section 2.7. Additionally, it is also possible to develop approximate constellation performance metrics as a substitute for simulation. This is explored further in Section 3.1.2. Finally, it is important to note that for many operational imaging satellites, a sun-synchronous orbit offers consistent lighting conditions for the images and maximizes the usable operation time of the satellite. While these orbits were not explored in this effort, it may be a worthwhile extension in future work. The RECON framework is still applicable, but the altitudes and inclinations of the satellites must be chosen more precisely.

2.4.2 Regional Observing Mode Constellation

The objective of the Regional Observing Mode (ROM) constellation is to focus coverage on one specific target region. One method that accomplishes this high coverage level uses

repeating ground track (RGT) orbits. Repeating ground track orbits have the property that the pattern of coverage on the Earth repeats after a finite period of time. Repeating ground track orbits can be used easily in conjunction with Walker constellations, as described in Section 2.4.3.

The ROM of the constellation features up to N satellites in circular orbits with repeating ground tracks that include the target location. This significantly increases the coverage of the target in comparison to the global observing mode. A RGT orbit is defined mathematically using the period of the satellite orbit and the period of Earth's rotation. Equation (2.5) gives the period of an orbiting satellite.

$$T_{SAT} = 2\pi\sqrt{\frac{a^3}{\mu}} \tag{2.5}$$

Here, a is the semi-major axis of the orbit ($R_E + h$ for a circular orbit) and μ is the gravitational parameter for the Earth, approximately $3.98 * 10^4 km^3 s^{-2}$. When the period of the satellite T_{SAT} divided by the Earth's period of rotation is an integer fraction, the satellite follows a RGT. The satellite returns to the same Earth longitude and latitude after a finite number of orbits, and repeats the identical path. Equation (2.6) describes this relationship:

$$\frac{T_{SAT}}{T_{EARTH}} = \frac{n}{k} \tag{2.6}$$

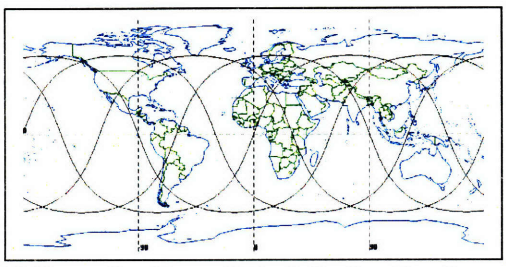
where k and n are integers, and T_{EARTH} is the length of a sidereal day, approximately 86,200 s. When $n = 1$, the ground track repeats after one day. For one day RGTs, k may be anywhere from one (geosynchronous) to sixteen (very low altitude). Table 2.1 lists the one day RGT altitudes. In general, the precise altitude of the satellite is determined based on k and n, and an infinite number of combinations exist.

An example RGT for $k = 6$ and $n = 1$ is shown in Figure 2-4(a). However, a small perturbation Δa to the satellite semi-major axis a generally results in a non-repeating ground track, since the period T_{SAT} has changed. After n days, the satellite does not return to the original starting longitude and latitude, but instead is displaced to the east or west. This can be seen in Figure 2-4(b). As a result, the satellite will, over time, sweep out all area bounded by the inclination of the orbit. These non-repeating ground tracks

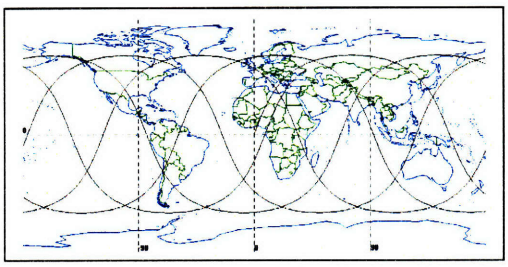
Table 2.1: List of RGT Altitudes for $n = 1$

k	Altitude h, km
1	35,786
2	20,184
...	...
11	2,147
12	1,666
13	1,248
14	881
15	554
16	262

are the foundation of the Walker constellation described in Section 2.4.1. The interaction between the repeating and non-repeating ground tracks is important to the RECON system architecture, and is discussed in the next section.



(a) Repeating Ground Track for $k = 6$ and $n = 1$.



(b) Non-Repeating Ground Track after Altitude Perturbation Δa.

Figure 2-4: Ground Tracks for Repeating and Non-Repeating Orbits

The predictability of the RGT orbits enables accurate calculation of coverage metrics. Each satellite is able to view a distance λR_E in all directions from its current nadir position on the ground track. This is the length of half of the footprint arc shown in Figure 2-1. One pass out of k travels over the target location. The time in view of the target during this pass is given by $\frac{2\lambda}{2\pi} T_{SAT}$. Equation (2.7) gives the fraction of time in view per n days.

$$\omega_{R,1} = \frac{\lambda}{\pi} \frac{T_{SAT}}{nT_{EARTH}} = \frac{\lambda}{\pi k} \tag{2.7}$$

While in ROM, up to N satellites are in RGT orbits that observe the target. Equation

(2.8) gives the percentage of access time per day using all N satellites.

$$\omega_R = 100\frac{N\lambda}{\pi k} \tag{2.8}$$

Th metric ω_R value may exceed 100% if a sufficient number of satellites are on orbit. While $\omega_R = 100\%$ would constitute continuous coverage of the target, $\omega_R > 100\%$ would indicate simultaneous coverage of the target by one or more satellites.

Under certain geometries, it is possible for a single satellite to image the target on two separate occasions within n days. This occurs when the target lays near the intersection of an ascending (South to North) and descending (North to South) pass of the RGT. In general, the number of intersection (node) locations depends on the parameters k and n. The two imaging opportunities correspond to the two passes of the target through the plane defined by the satellite's orbit. As a result, the satellite traversal of the intersecting ascending and descending RGT passes is separated in time by approximately one half day for $n = 1$. For an optical satellite sensor, this may render one of the two passes useless due to nighttime lighting conditions. Only certain configurations - where the satellite orbit plane aligns with the Earth day-night terminator line - enable the possibility of taking images on both ascending and descending passes. In this case the images are taken in the very early morning and late evening. Furthermore, these opportunities for increased target coverage are limited to the latitudes at which the nodes are located. For these reasons, the coverage effects associated with the RGT nodes are not considered.

Finally, this analysis does not account for perturbations on the satellite. Specifically, the J_2 perturbation causes the Right Ascension of the Ascending Node (RAAN) of the satellite to shift over time, resulting in misalignment of the orbit as designed in Equation (2.6). Orbital parameters of the satellite constellation can be adjusted to compensate for this effect. Equation (2.6) is modified as follows:

$$n\left(\dot{\theta_E} + \dot{\Omega}\right) Day_s = kT_{SAT} \tag{2.9}$$

$$\dot{\Omega} = -\frac{3}{2}\sqrt{\frac{\mu}{a^3}}J_2\left(\frac{R_E}{a(1-e)^2}\right)^2 cos(i) \tag{2.10}$$

where, $\dot{\theta_E}$ is the rotation rate of the Earth, which is 2π divided by the length of a sidereal day. The parameter e is the eccentricity of the orbit, and Day_s is the length of a sidereal

day in seconds. The drift of the longitude of the ascending node resulting from the J_2 perturbation is approximated by $\dot{\Omega}$. The approximate value of the second Earth zonal harmonic coefficient, J_2, is 0.001083. Higher order gravitational perturbations, and effects such as solar pressure and drag must be corrected as necessary using the satellite's on board propulsion system.

2.4.3 Constellation Reconfiguration

Section 2.4.1 described that the Walker constellation will be placed at a non-repeating ground track (NRGT) altitude to enable global coverage within the inclination band. The specific altitude of the satellites in the constellation determines the rate at which they sweep out coverage on the Earth. Altitudes closest to the RGT altitude take the longest to sweep out global coverage.

Consider circular orbits **A** and **B**, with periods T_A and T_B. Orbit **A** is a RGT orbit with semi-major axis a, calculated with Equations (2.5) and (2.6). Orbit **B** is a NRGT orbit with semi-major axis $a + \Delta a$. Note that since $a = R_E + h$ for circular orbits, any perturbation to a is directly reflected in h, thus $\Delta a = \Delta h$. Equation (2.11) describes the difference in period between the two orbits.

$$\Delta T = 2\pi \left(\sqrt{\frac{(a+\Delta a)^3}{\mu}} - \sqrt{\frac{a^3}{\mu}} \right) = \frac{2\pi}{\sqrt{\mu}} \left((a+\Delta a)^{3/2} - a^{3/2} \right) \tag{2.11}$$

The parameter μ is the gravitational parameter for the Earth. This equation can be linearized for small Δa by taking the derivative of Equation (2.5):

$$\Delta T = \frac{dT}{da}\Delta a = \frac{2\pi}{2}\sqrt{\frac{\mu}{a^3}}\frac{3a^2}{\mu}\Delta a = 3\pi\sqrt{\frac{a}{\mu}}\Delta a \tag{2.12}$$

The distance that the ground track deviates from the nominal repeating ground track when crossing the equator, per orbit, is given by:

$$\Delta d = \dot{\theta_E} R_E \Delta T = \frac{2\pi}{Day_s} R_E \Delta T \tag{2.13}$$

Equation (2.14) gives the relative ground track motion per k orbits.

$$\Delta D = k\Delta d = k\frac{6\pi^2 R_E}{Day_s}\sqrt{\frac{a}{\mu}}\Delta a \tag{2.14}$$

The parameter ΔD represents the relative distance that the ground track slips at the equator per n days. Figure 2-5 illustrates Δd and ΔD. Equation (2.14) is used in this thesis to describe deviations from a RGT orbit with semi-major axis a. However, this equation is valid for any two orbits whose altitudes are in close proximity.

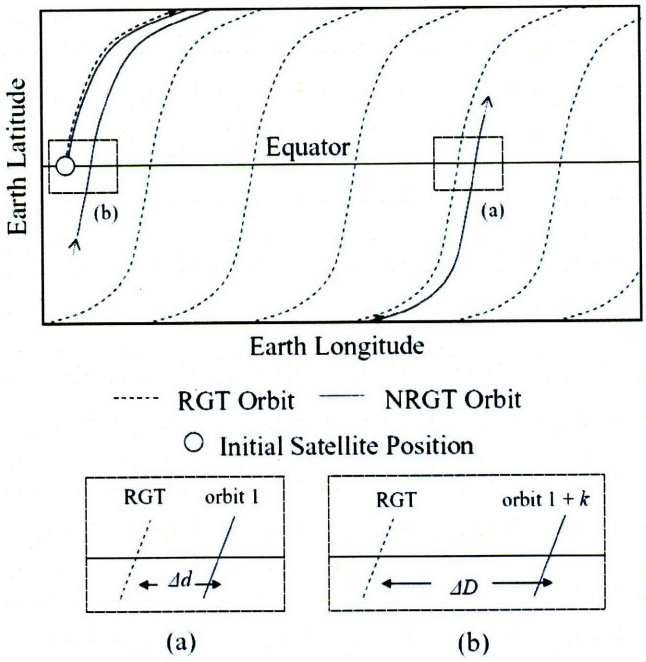


Figure 2-5: Illustration of Parameters Δd and ΔD

Note that Δa can be a positive or negative quantity. If Δa is positive, then the satellite will be at a higher altitude orbit than the corresponding repeating ground track. This causes the orbital period to be slightly longer. The Earth will complete n rotations before the satellite arrives back at the starting point. Thus, the ground track will slip to the west in longitude. Likewise, if Δa is negative, the orbital period will be shorter, and the satellite will arrive at the starting point before the Earth completes n rotations. The ground track will slip to the east.

The latitude and longitude of a target of interest uniquely identify the RGT that travels through the target location. Upon target notification, the proper timing at which to maneuver each satellite into ROM is determined. The worst case (longest) time is an important

parameter of the system design. If the Walker constellation is distributed in the traditional fashion, and a target arises immediately after the GOM constellation drifts through the appropriate ground track, then the constellation must be allowed to continue to drift until the next proper alignment occurs. Equation (2.15) gives the maximum distance that the satellite must traverse, along the equator, between two consecutive crossings of the RGT.

$$\Delta D_{max} = \frac{2\pi R_E}{k} \tag{2.15}$$

This distance is equivalently the maximum distance that the ground track of a NRGT satellite must slip before proper RGT alignment occurs and the reconfiguration maneuver can be performed. Equation (2.16) gives the drift time in days for a single satellite.

$$\omega_t = \frac{D_{max}}{\Delta D} = \frac{Day_s}{3\pi k^2}\sqrt{\frac{\mu}{a}}\frac{1}{\Delta a} \tag{2.16}$$

By launching a constellation of N satellites, the maximum distance of ground track slip decreases in proportion to N. The drift time decreases accordingly:

$$\omega_{t,G} = \frac{\omega_t}{N} = \frac{D_{max}}{\Delta D} = \frac{Day_s}{3\pi k^2}\sqrt{\frac{\mu}{a}}\frac{1}{\Delta a N} \tag{2.17}$$

Equations (2.16) and (2.17) do not consider the time spent on-orbit in ROM, $\omega_{t,R}$, before the first image is taken. The response time, as defined in Section 2.3, is approximately $\omega_{t,G} + \omega_{t,R}$. The maximum time required for reconfiguration of the entire constellation is based on the worst case location of a single satellite. This may be up to ω_t days.

2.4.4 Reconfiguration ΔV

At a large semi-major axis offset, Δa, the Walker constellation sweeps out global coverage rapidly. This property reduces the response time of the constellation. However, large altitude offsets require large ΔV to maneuver the satellite between the two configurations.

The relationship for a simple two-burn Hohmann transfer maneuver between two co-

planar circular orbits is shown in Equation (2.18).

$$\Delta V = \sqrt{\mu}\left[\left|\sqrt{\frac{2}{a_A} - \frac{1}{a_{tx}}} - \sqrt{\frac{1}{a_A}}\right| + \left|\sqrt{\frac{2}{a_B} - \frac{1}{a_{tx}}} - \sqrt{\frac{1}{a_B}}\right|\right] \tag{2.18}$$

$$a_{tx} = \frac{a_A + a_B}{2} \tag{2.19}$$

Here, a_A and a_B are the semi-major axes of orbits **A** and **B**, respectively. The semi-major axis of the transfer orbit is given by a_{tx}. For a small change in semi-major axis $\Delta a = a_A - a_B$, Equation (2.18) can be approximated by Equation (2.20).

$$\Delta V = \frac{1}{2a}\sqrt{\frac{\mu}{a}}\Delta a \tag{2.20}$$

Due to the $1/r^2$ effect of gravity, for the same magnitude change in semi-major axis, the maneuver requires less ΔV as altitude increases. Over the range of altitudes that support high-resolution Earth imaging (approximately 400 to 1000 km), the required ΔV for a given altitude change varies by nearly 10%. For an example constellation with $N = 20$ satellites at an altitude of 554km, Figure 2-6 shows the trade between response time (Equation (2.17)) and reconfiguration ΔV (Equation (2.20)). The altitude offset Δa is an implicit variable, ranging up to $\Delta a = 50$ km at $\Delta V = 25$ m/s.

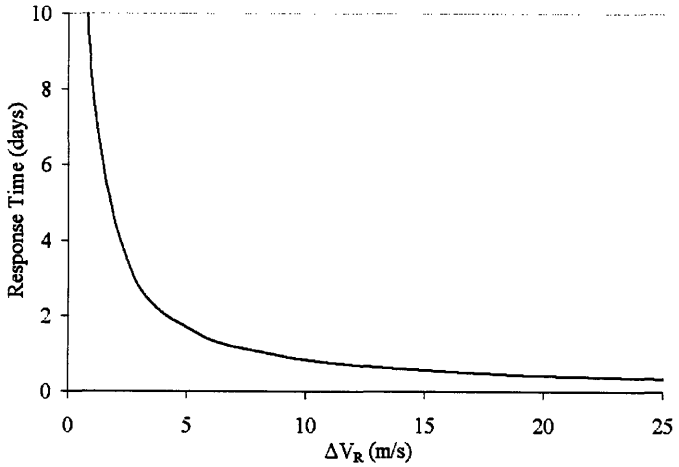


Figure 2-6: Responsiveness Fuel Requirements for $N = 20$ Satellites

As the ground track of a satellite in GOM gradually shifts across the Earth's longitude, it approaches the repeating ground track. At the appropriate time, the satellite executes a two burn Hohmann transfer maneuver to transfer into the ROM constellation. Mission

ground software can precisely time this maneuver by constructing a two-point boundary value problem using the orbital dynamics equations and an initial and final position and velocity appropriate for the GOM and ROM orbits.

2.4.5 Elliptical Orbits

One of the main design variables for the mission architecture is the orbit selection. Circular orbits offer the most predictable coverage results, and also simplify the re-configuration process. Elliptical orbits offer several benefits. The main benefit is increased access time over the region of interest. By placing the apogee of the repeating ground track orbit over the target, the percentage of on orbit time with the target in view will be greater than for any other location on the Earth. Consider Equation (2.5). The semi-major axis, a represents the average of the apogee and perigee radii of the orbit. Given a circular orbit at radius a, it is feasible to adjust r_a and r_p, the radii at apogee and perigee, respectively, to create an orbit with the same period as the original circular orbit. The satellite velocity at apogee decreases, thus distributing more of the time on orbit to the region near the target.

Equation (2.21) approximates the fractional increase in access time for an elliptical orbit over a circular orbit, where $r_a = r_p$.

$$\sigma \approx \frac{V_{circ} - V_a}{V_{circ}} = 1 - \sqrt{1-e} \tag{2.21}$$

Here, V_a is the velocity at apogee of an orbit with eccentricity e, and V_{circ} is the orbital velocity of a circular orbit, where both have the same semi-major axis, a. Figure 2-7 shows the relationship for a series of orbits which all have one day ($n = 1$) repeating ground tracks with $k = 14$. The semi-major axis is held constant at a value defined by Equations (2.6) and (2.5). The circular orbit corresponding to this semi-major axis has an altitude of 881 km. Appropriate balancing of the apogee and perigee altitude ensures that the period of the eccentric orbit remains constant.

The function described by Equation (2.21) is nearly linear in the range from zero up to $e = 0.09$. At this value of e, the apogee altitude is approximately 1450 km and the perigee altitude is 300 km - the lower practical bound due to drag. Even at this extreme, the dwell time for a target under the orbit apogee only increases by four percent. Additionally, due to the increased altitude, the optical system requires a significantly larger mirror diameter.

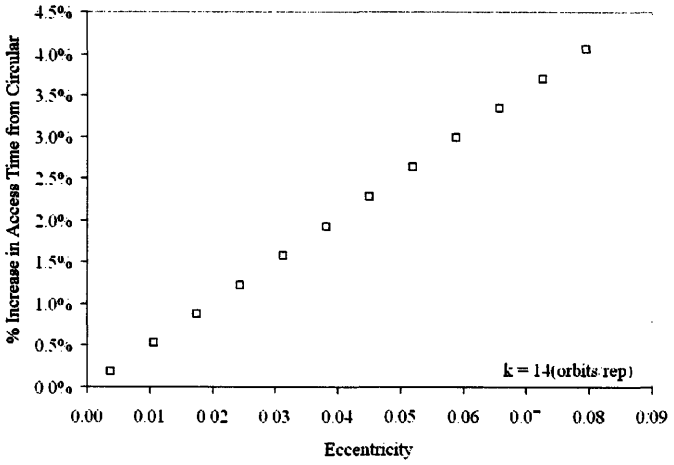


Figure 2-7: Access Time for Modified Eccentric Orbits

The associated costs quickly offset the dwell time benefits.

The RECON architecture may feature eccentric orbits as the foundation for either of the observing modes. Eccentric orbits, however, create more complicated timing constraints for the reconfiguration maneuvers. The use of eccentric orbits in global observing mode, while a possibility, may result in an unbalanced distribution of imagery. If the assumption is made that the satellites are designed to image at the apogee altitude, then part of the orbit may be unusable. For this assumption to be relaxed, the satellites must be designed to account for a significant dynamic range (h') to the target. It may be possible to design a constellation where apogee locations of the orbits are distributed over a range of latitudes and longitudes to provide more uniform coverage. This extension is not explored in this thesis.

2.4.6 Satellite System

For the purpose of this analysis, each satellite features an optical sensor, capable of imaging the Earth's surface at a required 1 m resolution. Necessary specifications for an optical system include the telescope type, focal length, pixel size, CCD array size, and aperture diameter. These details, however, may be abstracted into a more simple physics-based relationship. One major limitation of an optical system is the aperture diameter. The aperture diameter places a limit on the minimum angular resolution of the system. This is the diffraction limit, and is assumed binding when there are no atmospheric disturbances and mirror surfaces are smooth.

The angular resolution for a diffraction limited optical system is shown in Equation (2.22).

$$\theta_{res} = \frac{1.22\lambda_0}{D} \tag{2.22}$$

Here, λ_0 is the desired observing wavelength of the system. In this case, λ_0 is chosen as the largest wavelength to be observed. In the visible spectrum, this is set as $\lambda_0 = 750$ nm. The variable D is the aperture diameter, and θ_{res} is the angular resolution of the system. Equation (2.23) calculates the ground resolution of the system [38].

$$\Delta x = \frac{1.22\lambda_0 h'}{\hat{T}' \cdot \hat{h}' D} \tag{2.23}$$

where $\hat{h}'$ is the vector to the target, $\hat{T}'$ is the target surface normal vector, θ is the maximum off-nadir pointing angle of the satellite, and Δx is the ground resolution obtained by the optics. The value $\hat{T}'$, for a flat surface on the Earth, may be taken to be the Earth surface normal vector, $\hat{R}'$. Images taken at large off-nadir angles have poor resolution in the desired imaging direction, since $\hat{T}' \cdot \hat{h}'$ is small. However, if it is assumed that the image is taken of a 3-dimensional object, then $\hat{T}'$ can be assumed to have a component in the direction of $\hat{h}'$. Under this assumption, $\hat{T}' \cdot \hat{h}' \approx 1$. The required diameter, as a function of wavelength, range to target, and required resolution is described by Equation (2.24).

$$D = \frac{1.22\lambda_0 h'}{\Delta x} \tag{2.24}$$

The satellite bus for the RECON architecture is comparable to existing systems such as IKONOS and WorldView-1 [31, 69]. Spacecraft subsystems such as the Attitude Control System (ACS) for slewing are assumed part of a general spacecraft mass, and not modeled in detail in this effort. We have developed relationships for performance in ROM and GOM, as well as reconfiguration requirements. A cost model for the satellite constellation enables objective comparison between different RECON systems.

2.5 Cost Modeling

The architecture defined in Section 2.4 is analyzed using a set of performance metrics. The development of a cost model enables objective comparison of different constellations. The model features several modules which each describe elements of the constellation. Assume that the program budget for a RECON system is P[\$]. The primary objective of the model is to trade design parameters in order to generate a set of different designs which all meet a given cost budget. Constellation parameters such as number of satellites (N), optics size (D), and design altitude (h) influence the total cost, and each is discussed in detail in the following sections.

2.5.1 Payload

The cost model for the optical system is a function of the diameter of the aperture, which is calculated using Equation (2.24). Historical data suggest that cost is given by the relationship shown in Equation (2.25) [46].

$$C_{optics} = c_{optics} D^{2.6} \tag{2.25}$$

The parameter c_{optics} is chosen again based on data from the IKONOS spacecraft. IKONOS is an optical Earth-observing satellite in LEO, and serves as an excellent analog for a RECON vehicle.

2.5.2 Spacecraft Bus

An approximate relationship for the dry mass of a single satellite uses a scaling factor based on a benchmark satellite and the desired aperture diameter. The assumption is that the satellite bus dry mass scales with the payload diameter [38]. Equation (2.26) calculates m_{dry}, the dry mass of a RECON satellite.

$$\begin{aligned} m_{dry} &= qR^3 m_{dry,I} \\ R &= \frac{D}{D_I} \end{aligned} \tag{2.26}$$

The parameters D_I and $m_{dry,I}$ are reference satellite parameters. IKONOS serves as the benchmark satellite, with a dry mass of $m_{dry,I} = 653.6$ kg and aperture diameter $D_I = 0.7$

m. The parameter q is a scaling factor equal to 1 for $R > 0.5$ and equal to 2 for $R < 0.5$. Due to the ground resolution requirement of 1 m, the RECON architecture necessarily features satellites with $D > 0.35$ m. Thus, q can be fixed at a value of one. A lower bound on the mass, $m_{dry} > 200$ kg, eliminates trivially small spacecraft.

A linear model describes the spacecraft cost. Equation (2.27) gives the relationship.

$$C_{S/C} = c_{dry} m_{dry} \tag{2.27}$$

The parameter $c_{dry} = 0.113$ \$M/kg represents an estimate for small satellites [6].

2.5.3 Propellant Mass

The mission reconfiguration, station keeping, and disposal ΔV determine the required propellant mass. Equation (2.20) gives the ΔV_R for a single RECON satellite reconfiguration. The station-keeping ΔV_S requirements are a function of the estimated drag coefficient of the satellite and the atmospheric properties at the design altitude. Equations (2.28) and (2.29) describe the drag make-up requirements per orbit and per year, respectively.ermine the required propellant mass. Equation (2.20) gives the ΔV_R for a single RECON satellite reconfiguration. The station-keeping ΔV_S requirements are a function of the estimated drag coefficient of the satellite and the atmospheric properties at the design altitude. Equations (2.28) and (2.29) describe the drag make-up requirements per orbit and per year, respectively.

$$\Delta V_{s,ORBIT} = \pi \frac{C_D A}{m} \rho a \sqrt{\frac{\mu}{a}} \tag{2.28}$$

$$\Delta V_s = \frac{\Delta V_{s,ORBIT}}{T_{SAT}} year = \frac{1}{2} \frac{C_D A}{m} \frac{\rho \mu}{a} year \tag{2.29}$$

$$year = 60 * 60 * 24 * 365 \tag{2.30}$$

Here, T_{SAT} is the period of the satellite orbit in Equation (2.5). The term $\frac{C_D A}{m}$ is the ballistic coefficient of the satellite, and ρ is the atmospheric density at the satellite altitude. The ΔV requirement for end-of-life disposal is calculated using Equation (2.31), with $r_B =$

$r_{disp} = R_E + 50$ km, corresponding to the altitude of the disposal orbit.

$$\Delta V_d = \sqrt{\mu}\left|\sqrt{\frac{2}{a} - \frac{1}{a_{tx}}} - \sqrt{\frac{1}{a}}\right| \qquad a_{tx} = \frac{a + r_{disp}}{2} \tag{2.31}$$

The total ΔV_T needed on-board the spacecraft is a combination of the mission required ΔV_m (reconfiguration and station keeping) shown in Equation (2.32a), and the disposal ΔV_d.

$$\Delta V_m = 2N_R L_m \Delta V_R + \Delta V_s L_m \tag{2.32a}$$

$$\Delta V_T = \Delta V_m + \Delta V_d \tag{2.32b}$$

Here, N_R represents the number of reconfigurations per year, and L_m is the mission lifetime in years. Table 2.2 shows a sample ΔV budget for a ten year mission with two reconfigurations per year ($N_R = 2, L_m = 10$). The selected repeating ground track altitude is 554 km, and the Walker constellation altitude, with Δa = -20 km, is 534 km.

Table 2.2: Sample RECON ΔV Budget

ΔV **Requirement**	ΔV **Required**		**Frequency**
Reconfiguration burn	11	m/s	2 / reconfig.
Station-keeping	1	m/s	/ year
Disposal	144	m/s	Once
Total ΔV	593	m/s	Mission Lifetime
Propellant Mass	89	kg	Mission Lifetime

Equation (2.33) describes the propellant mass requirements. The mass needed for normal operations and disposal is calculated separately, based on the approximate total mass of the satellite at the time of the maneuver.

$$m_d = m_{dry}(e^{\frac{\Delta V_d}{V_0}} - 1) \tag{2.33}$$

$$m_p = (m_{dry} + m_d)(e^{\frac{\Delta V_m}{V_0}} - 1) \tag{2.34}$$

The parameter $V_0 = I_{sp}g$ is a characteristic of the propulsion system, where I_{sp} is the specific impulse of the propellant used and g is gravitational acceleration. With an estimated 400

kg dry mass, and required total ΔV_T of 593 m/s , the mission requires 89 kg of propellant. The propellant is assumed to be a liquid propellant similar to hydrazine, with $I_{sp} = 300$ s.

2.5.4 Economies of Scale

One of the benefits of building many satellites is the cost savings obtained through mass production [38]. This is common to any large scale manufacturing process, and important in the construction of large constellations such as Iridium and NAVSTAR (GPS). Having designed and built a single vehicle, the manufacture of a second can be completed at a fraction of the cost. The incremental cost for building the tenth vehicle, for example, is less than the ninth. The total cost of building N satellites is given as:

$$C = C_{first} L \tag{2.35}$$

where C_{first} is the cost of building the first satellite in Equation (2.36):

$$C_{first} = C_{optics} + C_{S/C} \tag{2.36}$$

with C_{optics} and $C_{S/C}$ from Equations (2.25) and (2.27), respectively. The parameter L is given by Equation (2.37):

$$\begin{aligned} L &= N^B \\ B &= 1 - log_2 \frac{1}{S} \end{aligned} \tag{2.37}$$

The value of S is based on N. A suggested model is that S is equal to 95% if $N < 10$, and S is equal to 90% if $10 \leq N < 50$. For $N > 50$, a value of S = 85% may be used. In the numerical implementation of this formula, a single logarithmic curve representation elimitates the step effect at the boundaries and thereby create a more smooth (and more realistic) model without local minima.

$$S = 100 * (1.0048 - 0.0432 * log(N)) \tag{2.38}$$

2.5.5 Launch

Launch costs comprise a large fraction of the cost of any space system. The optimization of satellite placement aboard various launch vehicles is complicated, and outside the scope of this work [33]. The launch cost model is simply a flat cost per kilogram plus additional upper stage costs for orbit injection. The cost per kilogram is the average of a series of U.S.-based launch vehicles, shown in Table 2.3.

Table 2.3: Estimated Launch Costs to LEO for U.S.-based Vehicles [38]

Launch Vehicle	Max Payload (kg)	Cost/kg (FY2000$K/kg)
Atlas II	6580	12.2 - 13.7
Atlas II A	7280	11.7 - 13
Atlas II AS	8640	11.6 - 12.7
Athena I	800	22.5
Athena II	1950	13.3
Athena III	3650	8.5
Taurus	1400	14.3 - 15.7

The average cost per kilogram for these launch vehicles is c_l = FY 2006 \$15,280 / kg. Additionally, the model includes costs for the use of an upper stage to maneuver the satellite into its mission orbit from the nominal parking orbit of $r_{inj} = R_E + 185$ km. Equation (2.18) determines the ΔV_{inj} needed to raise the orbit to the mission altitude. The mass of propellant needed for orbit injection is:

$$m_{inj} = (m_{dry} + m_p + m_d)(e^{\frac{\Delta V_{inj}}{V_0}} - 1) \tag{2.39}$$

with an $I_{sp} = 300$ s system. An additional 15% mass, an average of several legacy systems, represents engine infrastructure [38]. This total mass is then added to the satellite mass when calculating launch costs. Equation (2.40) outlines the total launch costs.

$$C_{launch} = c_l(m_{dry} + m_p + m_d + c_p m_{inj}) \tag{2.40}$$

2.5.6 Mission Operations

Mission operations account for a substantial portion of the cost of a large constellation. The operations costs for a constellation can be estimated as a faction of the first satellite cost.

For a traditional high-cost mission, the cost of operations per year is on average $c_{ops} = 3\%$ of the first unit cost [38]. The total mission operations cost is the product of this value and the mission duration. Equation (2.41) describes the operations costs per satellite.

$$C_{ops} = c_{ops} L_m C_{first} \tag{2.41}$$

2.5.7 Summary

The total constellation cost is the sum of satellite development and manufacture, operation, and launch costs.

$$C_{TOT} = L(C_{first}) + N(C_{ops} + C_{launch}) \tag{2.42}$$

Table 2.4 summarizes all of the constants used in this analysis. The simple model developed in this section will help provide insight into the types of systems that are best suited for the RECON architecture. Section 2.6 utilizes this model to generate different satellite constellations, and Section 2.7 explores the performance of these constellations.

2.6 Equal Cost Comparison

The cost model developed in Section 2.5 enables generation of a series of equal cost constellations. Various design attributes of the constellation, for example altitude and number of satellites, can be traded in order to find different constellations that meet the program budget P.

$$C_{TOT} \leq P \tag{2.43}$$

The ground resolution requirement requirement is 1 m at the limiting off-nadir angle. The number of satellites and the RGT altitudes are discrete quantities that have many thousands of combinations. An algorithm that efficiently distributes the project budget to various attributes enables generation of a set of equal cost constellations.

The predetermined budget of P dollars is an input to the algorithm. Starting at $k = 16$, the algorithm reduces k in integer values. At each altitude (value of k), the cost model estimates the cost of an integer number (N) satellites, where N starts at 1 and increases

Table 2.4: RECON Cost Model Parameters

Parameter	Value	Units	Description
Cost Parameters			
c_{optics}	15.2	$\$/m^{2.6}$	Mirror Cost
D_I	0.7	m	Reference Mirror Diameter (IKONOS)
$m_{dry,I}$	653.4	kg	Reference Dry Mass (IKONOS)
c_{dry}	0.113	\$M/kg	Dry Mass
c_l	15,280	\$/kg	Launch Cost
c_p	1.15	(ND)	Launch Upper Stage Infrastructure
c_{ops}	0.03	(ND)	Mission Operations Cost Fraction
Selected Mission Parameters			
r_{inj}	$R_E + 185$	km	Parking Orbit Altitude
r_{disp}	$R_E + 50$	km	Disposal Altitude
$\frac{C_D A}{m}$	200	kg/m^2	Ballistic Coefficient
I_{sp}	300	s	Thruster Specific Impulse
L_m	10	years	Mission Lifetime
N_R	2	/year	Number of Reconfigurations per Year
Δx	1	m	Ground Resolution Requirement
Constants			
λ_0	750	nm	Observing Wavelength
μ	3.98e5	km^3/s^2	Earth Gravitational Constant
R_E	6,378	km	Earth Radius
g	9.8	m/s^2	Gravitational Acceleration
Day_s	86,200	s	Sidereal Day

until the architecture exceeds the budget. In the case that the budget is insufficient when $N = 1$, the current altitude and all higher altitudes can be eliminated. At each choice of N, a bound and search algorithm calculates the maximum angle of regard, θ, for the constellation. The algorithm first determines if the choice of N is feasible by seeing if the ground resolution requirement can be met for $\theta = 0$. If N satellites are feasible, the value of θ is increased until the baseline cost is exceeded, or the limiting angle ($\theta = \rho$ in Figure 2-1) is reached. A binary search executes to find the maximum angle of regard θ for N satellites under the cost constraints. The process is repeated to attempt to distribute P over $N+1$ satellites. For very low altitudes, the algorithm limits the number of satellites, since with a sufficiently high budget P, an arbitrarily large number of satellites can be built. Such a large number of satellites is difficult to simulate, and is not pursed in this chapter. The total number of satellites is limited to $N = 45$. Chapter 3 develops estimates for GOM performance which allow consideration of larger constellations by alleviating simulation requirements.

For the purpose of generating a baseline cost, a constellation of 20 satellites in a 554 km circular orbit with an angle of regard of 30° was used. The cost model predicts a 0.6 m aperture diameter, and a first satellite unit cost of $49.6M. The dry mass of each satellite is 400 kg, and the propellant mass is 74 kg. The total launch cost for the 20 satellites is $160M, and the total mission cost is $728M. It should be emphasized that this mission cost is very likely an underestimate of the true cost - as modern satellite mission budgets often exceed $500M per satellite. This is discussed further in Section 3.4.1. However, the conclusions to be drawn rely on comparing different constellations that reflect the same cost model. While the absolute cost may be inaccurate, the relative costs provide a good picture of the design trade space. More accurate cost models can be integrated into this analysis as desired.

Figure 2-8 shows the design options given a total budget of $P = \$728M$ in the cost model.

Notice that the budget allows for the largest number of satellites at the lowest altitude. As altitude increases, fewer satellites can be purchased due to increasing optics requirements. For example, if we select $N = 1$ satellite, there are four different purchasing options. We can place the satellite at $h = 262$ km with an angle of regard $\theta = 74°$, at $h = 554$ km with $\theta = 60°$, at $h = 881$ km with $\theta = 48°$, or at $h = 1248$ km with $\theta = 25°$. Above 1248 km

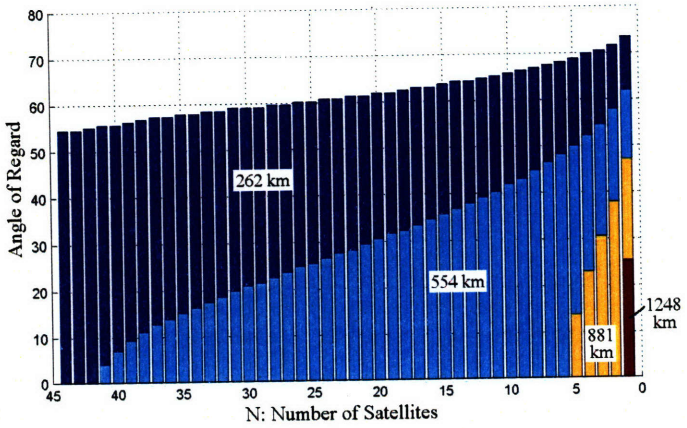


Figure 2-8: Equal Cost Constellation Options for $P = \$728\text{M}$

altitude, the cost of a single satellite featuring optics able to image to a resolution of 1 m exceeds the budget P. For a constellation of $N = 30$ satellites, the mission altitude can be either $h = 262$ km with $\theta = 59°$, or $h = 554$ km with $\theta = 21°$. Higher altitudes cannot support $N = 30$ satellites under the budget constraint.

The set of constellations shown in Figure 2-8 are evaluated in the next section. Since each can be built under the same budget P, the best performing constellations in terms of coverage are also the best in terms of overall value.

2.7 Performance Analysis

Performance analysis of each of the constellations generated in Section 2.6 provides insight into the trends in the Recon architecture. MATLAB® and Satellite Tool Kit (STK)™ programs allow for quick comparison of many Walker constellation designs. Analysis of the ROM repeating ground track orbits is done in MATLAB® using Equation (2.8), as the orbits' predictability obviates the need for simulation.

Three primary metrics, described in Section 2.3, are important when considering the performance of a RECON constellation. We define ω_G and ω_R as the global and regional mode coverage time metrics, respectively. The value ω_R is given by Equation (2.8), while ω_G is determined via simulation. The geometric mean of the performance values in global observation and regional observation modes is a metric that evaluates overall system per-

formance. Equation (2.44) shows this relationship.

$$\Omega = \sqrt{\omega_G \omega_R} \tag{2.44}$$

A representative subset of the 149 options shown in Figure 2-8 was selected for evaluation. A Δa of 20 km is assumed. Performance metrics were evaluated at 0° latitude, because, for a Walker constellation, 0° represents the worst cast coverage and revisit time performance relative to other latitudes. The longitude of the target is irrelevant due to the nature of the constellation ground track motion over time.

Simulations were run using a time period of 90 days with the two-body orbit propagator. The analysis can be repeated by modifying the design altitude to account for the J_2 oblateness perturbation, and simulating using the J_2 propagation model. The Sun constraint was left as inactive - images taken during the night are still counted. Activating the Sun constraint causes proportional performance 'losses' over all designs, but does not provide insight on the value of the designs themselves. Table 2.5 shows the simulation results. The revisit time shown is for the entire constellation.

The results are sorted in descending order using the performance metric Ω. This serves to help organize the information, but it is still important to analyze each performance metric. The highest ranking constellations by the Ω metric also have a low response time, low regional mode revisit time, and high access time in both modes. These are all desirable attributes. The model predicts that very low altitude constellations perform best over both observing modes.

It is likely that drag considerations at such a low altitude (262 km) will render this constellation very difficult to operate. The model includes station-keeping costs (fuel), but does not account for the additional infrastructure (fuel tanks, structural support) needed for the large propellant mass. Higher altitude constellations are not significantly affected by this assumption since the propellant mass is a smaller fraction of the total vehicle mass.

The best performing constellation, at the more reasonable 554 km altitude, is a system of 16 satellites each with a 34.4° angle of regard. This constellation achieves 1.01% per day (14.5 minutes) coverage of any target at the equator in GOM, with an average revisit time of 142 minutes. In ROM, the constellation achieves 2.06% per day (30 minutes) coverage, with a revisit time of 88 minutes.

Table 2.5: Equatorial Performance Data for Constellations of Equal Cost, $P = \$728M$, $N_R = 2/year$, $L_m = 10$

Constellation						GLOBAL		REGIONAL	
Rank	Ω	h **(km)**	N	θ (°)	**Resp. Time,** $\omega_{t,G}$ **(hr)**	**Access Time %/Day,** ω_G	**Revisit Time (hr)**	**Access Time %/Day** , ω_R	**Revisit Time (hr)**
1	3.51	262	40	55.6	0.35	2.46	0.88	5.00	0.57
2	3.21	262	60	47.0	0.23	1.91	0.82	5.39	0.38
3	2.75	262	70	41.3	0.20	1.48	0.86	5.10	0.32
4	2.68	262	20	61.9	0.69	2.16	1.33	3.32	1.16
5	2.06	262	80	33.8	0.17	0.96	1.01	4.42	0.29
6	1.85	262	10	65.9	1.39	1.66	2.17	2.07	2.34
7	1.44	554	16	34.4	0.96	1.01	2.36	2.06	1.47
8	1.41	554	20	30.4	0.77	0.91	2.22	2.19	1.17
9	1.38	554	10	41.8	1.54	1.11	2.84	1.71	2.35
10	1.36	262	90	25.2	0.15	0.53	1.29	3.48	0.26
11	1.32	554	24	26.4	0.64	0.78	2.20	2.22	0.98
12	1.07	554	30	20.6	0.51	0.55	2.33	2.09	0.78
13	0.47	881	2	37.8	8.66	0.44	10.70	0.51	11.91
14	0.40	262	100	10.9	0.14	0.10	2.85	1.57	0.24
15	0.37	881	4	22.9	4.33	0.25	10.12	0.54	5.95
16	0.25	554	40	6.9	0.39	0.07	5.52	0.89	0.59

The performance improvements observed are shown in Table 2.6, for the top ten architectures. The access time may improve by a factor of two to three, and the revisit time may decrease by up to 80% for a given reconfiguration. It is important to note that the improvements may be artifically large if the initial GOM has very poor performance. For example, consider the Rank 5 constellation at 262 km altitude. Because of the limited field of regard, the GOM performance is low. Transitioning to ROM provides great improvement, but does not necessarily indicate that the constellation is a good design overall.

Table 2.6: Reconfiguration Performance Improvements at the Equator

Constellation				Access Time %/Day			Revisit Time (hr)		
Rank	h **(km)**	N	θ (°)	**GOM**	**ROM**	**Impr.**	**GOM**	**ROM**	**Impr.**
1	262	40	55.6	2.46	5.00	104%	0.88	0.57	35%
2	262	60	47.0	1.91	5.39	182%	0.82	0.38	54%
3	262	70	41.3	1.48	5.10	245%	0.86	0.32	62%
4	262	20	61.9	2.16	3.32	54%	1.33	1.16	13%
5	262	80	33.8	0.96	4.42	358%	1.01	0.29	72%
6	262	10	65.9	1.66	2.07	25%	2.17	2.34	-8%
7	554	16	34.4	1.01	2.06	105%	2.36	1.47	38%
8	554	20	30.4	0.91	2.19	141%	2.22	1.17	47%
9	554	10	41.8	1.11	1.71	54%	2.84	2.35	17%
10	262	90	25.2	0.53	3.48	554%	1.29	0.26	80%

It is worth noting that such a direct comparison of coverage and revisit time parameters assumes constant image resolution. The resolution however varies considerably based on the target geometry. The required image resolution Δx is 1 m at the maximum off-nadir angle θ. Images taken at an off-nadir angle less than θ feature better resolution. The coverage and revisit time in GOM include imaging opportunities of any resolution below the $\Delta x = 1$ m requirement. The performance metrics in ROM, however, result from a RGT which passes directly over the target. This pattern consistently provides opportunity for high resolution imagery taken from directly overhead, with a significantly shorter range to target.

2.8 Summary

This chapter presented a model for a Responsive Space architecture that utilizes global and regional observation modes. A cost model was developed and analysis performed on a series

of constellations. The results indicate that low altitude constellations are best, driven by the optical requirements. Within each altitude there is an optimal constellation, shown in Table 2.7. Reconfiguration offers significant improvements in coverage of the region of interest. Due to the extreme station keeping requirements, the optimal design at the 262 km altitude is rendered impractical. This is discussed further in Section 3.3.

Table 2.7: RECON Optimal Designs for Each RGT Altitude

Altitude h, km	**262**	**554**	**881**
N	40	16	2
$\theta(deg)$	55.6	34.4	37.8
Resp. Time (days)	0.35	0.96	8.66
ω_G	2.46	1.01	0.44
ω_R	5.00	2.06	0.51
Transition Gains	103%	104%	16%

A limitation encountered in this analysis is the requirement for simulation of Walker orbits. This is a very time consuming aspect which limits the number of different systems that can be evaluated. Chapter 3 introduces approximate metrics for the global satellite coverage, and develops an optimization formulation to evaluate RECON system designs. Additionally, a responsive UAV system architecture is introduced to serve as a comparison in evaluating the cost-effectiveness of RECON.

2.9 Future Work

There are several areas for improvement of this research:

- Orbital Analysis: Consideration of higher order orbital perturbations, as well as further research into constellations where the individual orbital planes are placed at different inclinations.

- Cost Model: More detailed modeling of spacecraft subsystems will produce more realistic cost estimates. With these improvements, a more accurate set of constellations can be generated.

- Consideration of $n > 1$ RGT Orbits: This analysis was limited to RGT orbits that have a repeat time of one day. It was assumed that images of the target could only

be taken when the satellite was on the pass of the RGT that went through the target. With a large enough off-nadir angle θ, the target may be positioned between different passes of the RGT. This is more easily done for $n > 1$, where the spacing of adjacent RGT passes decreases. With this change, more flexibility is available in the designed ROM orbit altitude.

- Performance Across Latitude: Analysis of performance at high latitudes, near the constellation inclination.

- Response Time: Estimates were developed for the response time of a constellation based on the time required for the first satellite to enter ROM. The time spent on orbit while in ROM before imaging the target was neglected, but is in fact be up to $n = 1$ days. It may be possible to formulate an optimization problem for the reconfiguration process to develop better insights.

- On Orbit Fuel Depots: The fuel on board each satellite is a limited resource, and critical to the reconfiguration objective. The launch of fuel depot satellites could extend the lifetime and flexibility of the constellation. Recent research being done on autonomous docking in space may support these objectives [50].

Chapter 3

ReCon Optimization and UAV Alternatives

Chapter 2 introduced the RECON system architecture. A cost model was developed to support performance analysis of different RECON constellations. Simulations were run to determine the constellation performance and the results were analyzed. This chapter takes an alternate approach to analyzing the RECON architecture by introducing a global coverage performance metric to eliminate the need for time-simulation. Analysis is performed under this modified framework to explore the nature of the optimal constellations. The second part of this chapter introduces an alternate responsive architecture, using UAVs, and draws comparisons between the cost and performance of each system.

3.1 Modeling Recon as an Optimization Problem

Given the nature of the reconfigurable architecture described in Chapter 2, it is natural to consider modeling the RECON architecture as an optimization problem. Simulation in STK^{TM} is restrictive due to long computation times, and eliminating the need for this element improves the efficiency of the trade-space exploration. This chapter investigates an optimization approach using an alternative performance metric for global coverage.

3.1.1 Design Variable Constraints

The relationships explained in Chapter 2 are repeated here, split into categories of design variables and cost variables. The primary design variables are listed in Equations (3.1).

$$D = \frac{1.22\lambda_0 h'}{\Delta x} \tag{3.1a}$$

$$m_{dry} = c_{dry} D^3 \tag{3.1b}$$

$$\Delta V_R = \frac{1}{a}\sqrt{\frac{\mu}{a}}\Delta a \tag{3.1c}$$

$$\Delta V_T = 2N_R L_m \Delta V_R + \Delta V_s L_m + \Delta V_d \tag{3.1d}$$

$$m_d = m_{dry}(e^{\frac{\Delta V_d}{V_0}} - 1) \tag{3.1e}$$

$$m_p = (m_{dry} + m_d)(e^{\frac{\Delta V_m}{V_0}} - 1) \tag{3.1f}$$

$$m_{inj} = (m_{dry} + m_p + m_d)(e^{\frac{\Delta V_{inj}}{V_0}} - 1) \tag{3.1g}$$

The values ΔV_s, ΔV_d, and ΔV_{inj} are a function of the semi-major axis a and the disposal (insertion) altitude. Since they each take discrete values based on the chosen orbital altitude, they will be handled separately. This is described in Section 3.2. Equations (2.29), (2.31), and (2.39) govern these three parameters.

The four major contributions to the program cost are from the spacecraft payload, bus, launch, and operations costs. These contributions are described by Equations (3.2). Program development costs are included in the payload and bus costs.

$$C_{optics} = c_{optics} D^{2.6} \tag{3.2a}$$

$$C_{S/C} = c_{dry} m_{dry} \tag{3.2b}$$

$$C_{launch} = c_l(m_{dry} + m_p + m_d + c_p m_{inj}) \tag{3.2c}$$

$$C_{ops} = c_{ops}\left(C_{optics} + C_{S/C}\right) \tag{3.2d}$$

The major constraint of this design process, as described in Chapter 2, is the total program budget, P. Equation (2.42) is repeated here.

$$\begin{aligned} L &= AN^B \\ L(C_{optics} + C_{S/C}) + N(C_{launch} + C_{ops}) &\leq P \end{aligned} \tag{3.3}$$

Here, $L(N)$ is a function which represents the economy of scale savings when manufacturing many satellites. The function $L(N)$ is a curve fit to Equations (2.35) and (2.37), with parameters $A = 1.5025$ and $B = 0.6442$. The design constraints listed ensure that only feasible architectures are explored. In order to evaluate these architectures, several performance metrics are developed.

3.1.2 Performance Metrics

The desired performance properties of the satellite constellation are three-fold, as discussed previously. An optimal constellation will have high coverage in regional observing mode (ROM), high coverage in global observing mode (GOM), and a low response time. Three approximations are presented for these metrics, in order to circumvent the need for simulation in STK^{TM}.

The first metric is approximation of the average regional mode access time to a single target. This equation is derived in Section 2.4.2, but repeated here for completeness.

$$M_1 = 100\frac{N\lambda}{\pi k} \tag{3.4}$$

Metric M_1 is an analytical calculation based on predictable orbit mechanics, and is therefore an accurate representation of the problem.

The second important figure of merit for the constellation design is the performance in global observation mode. This is calculated based on the total footprint of the constellation over the globe. A constellation of N satellites, each with a footprint given by Equation (2.3), is distributed over the Earth. The surface area of the Earth is given by $4\pi R_E^2$, however, the constellation does not access all latitudes. The fraction of the Earth's surface area that is accessible by the constellation is given by $sin(i)$. The global coverage relationship may then be predicted in Equation (3.5) as the total constellation footprint divided by the observable surface area of the Earth.

$$\begin{aligned} M_2 &= 100\frac{N(2\pi R_E^2(1-cos(\lambda))\xi}{4\pi R_E^2 sin(i)} \\ &= 100N\frac{1}{2}\frac{1-cos(\lambda)}{sin(i)}\xi \end{aligned} \tag{3.5}$$

The metric M_2 gives the percent access time per day to any target on the Equator. There is

an additional factor ξ included because of the disparity between coverage at the equator and coverage near the inclination of the constellation. The metric is designed for coverage at the equator, which is the worst case. Analysis in STK^{TM} indicates that for $i = 60°$, the value of coverage at the equator is approximately 0.55 times the average value across all latitudes within the inclination band. The value of ξ, the ratio of Equatorial coverage to average coverage, is thus taken as approximately 0.55. See Appendix A for details regarding this parameter and validation of M_2. Metric M_2 is based on geometric arguments and accurately models the scenario. Comparison with simulations in STK^{TM} reveals that M_2 provides an error of approximately 3% on average. The standard deviation of the error is approximately 16.5%.

Finally, the third figure of merit is the response time of the constellation. The response time of the constellation is an analytical result from the expressions derived in Section 2.4.3 as Equation (2.17). Metric M_3 is given by Equation (3.6). The inverse of response time is used so that lower values of response time yield larger figure of merit values.

$$\begin{aligned} M_3 &= (ResponseTime)^{-1} \\ &= \left(\frac{2\pi R_E}{Nk}\frac{1}{\Delta D}\right)^{-1} \\ &= \frac{3\pi}{Day_s}\sqrt{\frac{a}{\mu}}k^2\Delta a N \end{aligned} \tag{3.6}$$

The response time does not include the time spent while on the repeating ground track orbit, before the first image of the target is taken. Given the potentially large field of regard of the satellites, it is likely that an image of the target can be captured by a satellite still in the Walker configuration, before a reconfigured satellite passes overhead in a repeating ground track. In either case, the responsiveness of the image strongly correlates with metric M_3.

3.1.3 Objective Function

The choice of an objective function for the problem is largely at the discretion of the system architect. An appropriate objective function will: 1) favor architectures with high figures of merit M_1, M_2, M_3, and 2) Promote a balance between M_1, M_2, M_3. In order to meet criteria 1), a weighting function α is employed to normalize the three metric values. The figure of merit M_3 has a nominal value near one, while M_1 and M_2, which represent the

percentage of coverage time per day in ROM and GOM, have nominal values near 2 and 1, respectively. In order to normalize the individual objectives, a function q_1 in Equation (3.7) is introduced.

$$q_1 = [\alpha_1 \; \alpha_2 \; \alpha_3] \cdot [M_1 \; M_2 \; M_3] \tag{3.7}$$

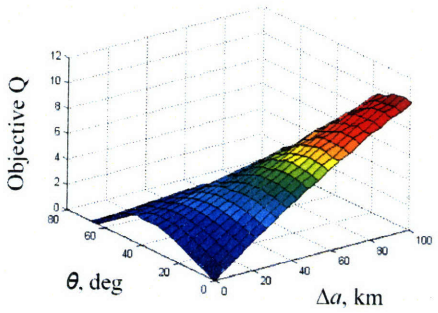
The metric values are normalized using $\alpha = [0.5 \; 1 \; 1]$. To meet criteria 2), the difference between metric values must also be considered in the objective function. This is accomplished by constructing q_2 in Equation (3.8).

$$q_2 = \beta \left(|\alpha_1 M_1 - \alpha_2 M_2| + |\alpha_2 M_2 - \alpha_3 M_3| \right) \tag{3.8}$$

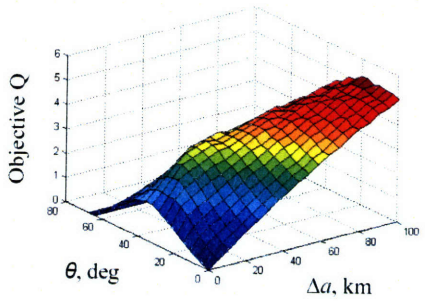
Here, β is a second weighting parameter used to adjust the contribution of the balancing objective to the overall objective function. Finally, the full objective function is constructed as in Equation (3.9).

$$\begin{aligned} Q &= q_1 - q_2 \\ &= \alpha \cdot \mathbf{M} - \beta \, |\alpha_1 M_1 - \alpha_2 M_2| - \beta \, |\alpha_2 M_2 - \alpha_3 M_3| \end{aligned} \tag{3.9}$$

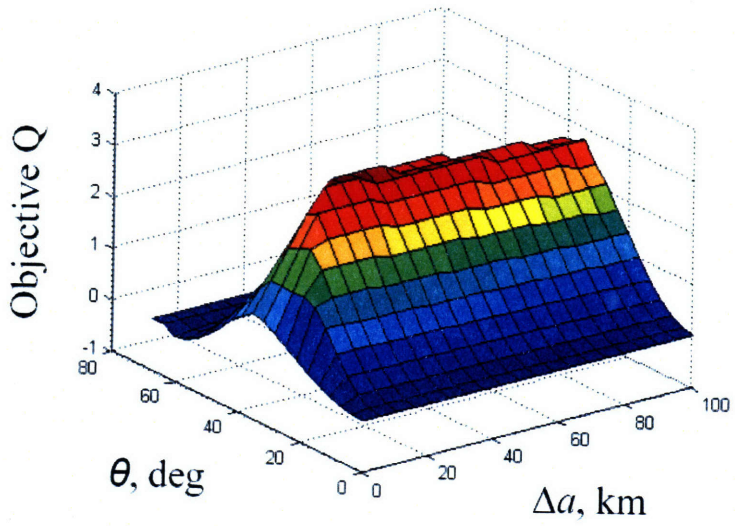
In the analysis to follow, β is set to a value of one. Low values of $\beta (< 0.5)$ imply little interest in balancing the attributes of the constellation. Optimized results for low values of β tend to favor one particular objective, for example minimizing the response time. High values of $\beta (> 1.5)$ imply that balancing constellation performance is most critical, eliminating flexibility in the nature of the solution. Figure 3-1 shows the transition from small to medium values of β. The tradespace shown is Δa versus θ for fixed altitude $h =$ 554 km with $k = 15$. The optimal solution for $\beta = 0$ is one in which the altitude offset Δa is maximized (thus response time is minimized), but the maximum off-nadir angle θ is left to be only a few degrees. This results in very short response time at the expense of ROM and GOM coverage. At $\beta = 1$, the optimal solution has a non-zero angle θ. The constellation specified by this optimum provides balanced ROM and GOM coverage, as well as low response time. A value of $\beta = 1$ suffices in providing a balance between the three objectives.



(a) $\beta = 0.$



(b) $\beta = 0.5$



(c) $\beta = 1.0$

Figure 3-1: Objective Function Evaluation for Increasing Values of β

3.2 Optimization Process

This section explores optimal solutions under the framework provided in Section 3.1. The general problem type is a Mixed Integer Non-linear Program (MINLP). Several of the constraints listed for this problem are non-linear, and the discrete choice of the repeating ground track altitude adds a combinatorial (i.e. integer) aspect to the problem. Methods exist for solving Mixed Integer Non-linear Programs, for example Benders Decomposition [16]. The discrete selection of altitude related parameters can be performed by adding the linear constraints described by Equations (3.10).

$$a = \sum_{i=1}^{T} a_i y_i \tag{3.10a}$$

$$k = \sum_{i=1}^{T} k_i y_i \tag{3.10b}$$

$$\Delta V_s = \sum_{i=1}^{T} \Delta V_{s,i} y_i \tag{3.10c}$$

$$\Delta V_d = \sum_{i=1}^{T} \Delta V_{d,i} y_i \tag{3.10d}$$

$$\Delta V_{inj} = \sum_{i=1}^{T} \Delta V_{inj,i} y_i \tag{3.10e}$$

$$\sum_{i=1}^{T} y_i = 1 \tag{3.10f}$$

$$y_i \in \{0, 1\} \tag{3.10g}$$

Here, T is the total number of orbit options available, beginning at the lowest altitude. For example, $a_i = 6640$ km, $k_i = 16$ represents the one-day repeating ground track orbit at a 262 km altitude, where $a_i = R_E + h_i$. The values $\Delta V_{s,i}$, $\Delta V_{d,i}$, and $\Delta V_{inj,i}$ can be calculated for each altitude a priori using the equations in Section 2.5.3 and 2.5.5.

Several factors encourage the use of full enumeration rather than optimization. First, the set of combinatorial options is limited to the number of feasible repeating ground track altitudes. For LEO altitudes with $n = 1$ day repeat periods, this is conservatively less than five, since $k < 12$ represents orbital altitudes above 1,666 km. Orbits higher than this will certainly experience prohibitive optics requirements to meet the 1 m ground resolution

objective. Additionally, the equality constraints listed in Equations (3.1) and (3.2) are largely uncoupled. Specifically, the selection of k, θ, and Δa determine the value of all of the constraints. Finally, the discovery of the global maximum during optimization requires that the objective function and constraints be convex. It is not immediately clear that all of the constraints are convex.

A set of modules were developed in MATLAB® to explore the design trade space. The process is carried out in the following steps:

1. Select an orbital altitude to fix k, h, a.

2. Determine the limiting off-nadir angle θ, based on Earth geometry.

3. Iterate over the range of Δa and θ.

   Calculate $D, m_{dry}, \Delta V_R, \Delta V_S, \Delta V_T, m_p$.

   Evaluate C_{optics}, $C_{S/C}$, C_{launch}, and C_{ops}.

   Solve Equations (3.3) for L and N.

   Evaluate the objective function and store the result.

4. Return the optimal function value and corresponding design.

Given the design of a single satellite, the number N that may be produced given the budget P is calculated. At each iteration the current design is compared to the discovered optimum design, and the new optimum is stored. This process allows for very quick analysis of different constellation configurations. The optimal constellation across different altitudes can be found simply by comparing the optimal constellations within each altitude. Appendix D contains the source code. The next section presents the optimal constellations for a budget comparable to that used in Chapter 2.

3.3 Analysis Results

Sections 3.1 through 3.2 described the process for quickly evaluating RECON constellations. This section presents the results of this approach using the same mission parameters as in Chapter 2. Figure 3-2 shows the optimization space across Δa and θ for a 554 km altitude orbit design.

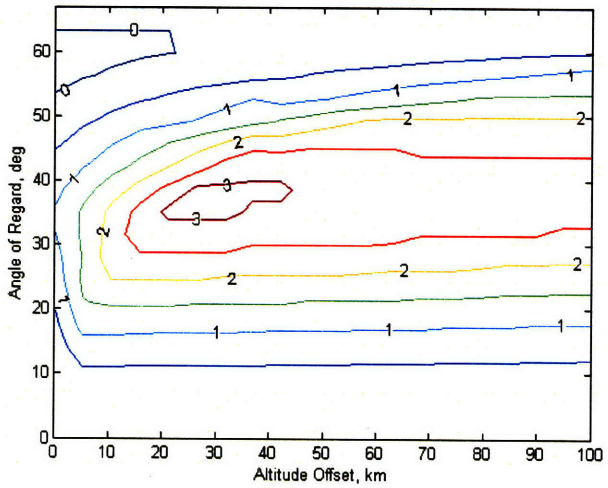


Figure 3-2: Contour Plot of RECON Optimization Space for 554 km Altitude Design

Table 3.1: Optimal RECON Designs at Three Design Altitudes

Parameter	**Constellation**		
	A	**B**	**C**
h, km	262	554	881
N	31	16	4
θ, deg	58.3	35.2	25.9
D, m	0.48	0.63	0.91
Δa, km	26.3	21.1	36.8
m_{dry}, kg	216	487	1441
$m_p + m_d$, kg	529	113	571
ΔV_R, m/s	15.4	11.5	18.8
ΔV_T, m/s	3642	616	983
Q	6.90	3.12	0.92
M_1	4.36	2.13	0.620
M_2	2.55	1.04	0.307
$1/M_3$	0.341	0.920	2.35

It is clear that an optimum exists between $\theta = 30°$ and $\theta = 45°$, and above $\Delta a = 20$ km. The optimal design for this altitude is shown in Table 3.1, alongside the results at two other design altitudes. There are several noteworthy trends. As expected, fewer satellites may be purchased at higher altitudes due to increasing optics size requirements. The optimal value of Δa is similar across all three altitudes. The objective value Q of the optimal constellations is highest at the lowest design altitude. As the design altitude increases, the optimal objective value decreases.

Table 3.2: Optimal RECON Design Comparison to Simulated Results

Parameter	Value
h, km	554
N	16
θ, deg	35.2
D, m	0.63
Δa, km	20
m_{dry}, kg	487
m_p, kg	109
ΔV_R, m/s	11.0
ΔV_T, m/s	593
Q	3.10
M_1	2.13
M_2	1.04
$1/M_3$	0.97

It is important to note that the optimal constellation design at 262 km features a propellant mass of 529 kg per satellite, approximately 71% of the satellite wet mass. Although the stationkeeping ΔV requirement was modeled in this effort, one shortcoming is the absence of additional structural mass (increased bus size, propellant tanks) due to the large amount of propellant. This effect is significant in cases where the propellant mass is a large fraction of the total mass. The results at higher altitudes are not appreciably affected, as the propellant mass fractions for designs B and C are a more reasonable at 19% and 28%, respectively.

In order to draw a comparison to the analysis in Chapter 2, the analysis is repeated for Δa limited to be less than 20 km. The result is shown in Table 3.2, and is very similar to constellation **B** in Table 3.1. As in Chapter 2, the optimal constellation at 554km altitude features $N = 16$ satellites, each with an angle of regard θ of approximately 35°. The coverage and response time properties are very similar, since they flow immediately from N and θ.

This section introduced an optimization approach to the RECON architecture design problem. By eliminating the need for simulation in STK^{TM}, the analysis times are reduced significantly, to the order of seconds instead of hours. The trends seen in the optimal solutions in Chapter 2 appear again under the optimization formulation. Low altitudes tend to perform the best. Optimal systems tend to have an angle of regard θ between 30°

and 40°, since above this the costs become prohibitive. The next section investigates the use of UAVs as an alternative to the RECON responsive architecture to draw performance and cost comparisons across two different domains.

3.4 Utility of UAVs as ReCon Alternative

There are many approaches for Responsive Space, as discussed in Section 2.2. In the interest of comparing RECON to another responsive architecture, this thesis studies the use of UAVs to perform reconnaissance. This section compares several candidate UAV architectures to RECON in terms of cost and capability.

UAVs are considerably more agile then spacecraft in terms of performing reconnaissance on a target. If several are held in storage for use in an emergency, they can be delivered to the appropriate site and deployed for continuous use. Alternatively, if many are stationed all over the globe, they can be very responsive to any arising event, possibly providing coverage within hours. Reconnaissance UAVs fly at altitudes between 6000 and 20000 m, so their capacity to take high resolution images is greatly increased over spacecraft which orbit at hundreds of kilometers in altitude. Additionally, with maximum mission flight duration over 24 hours, continuous monitoring of the target requires only several vehicles. These performance capabilities show promise for persistent surveillance. In order to support the feasibility of the RECON architecture, comparisons are drawn to other reconnaissance systems. For these reasons, this thesis explores UAVs as an alternative to satellites in performing this reconnaissance task.

3.4.1 ReCon Costing

One of the important elements of the RECON satellite constellation research was the development of the cost model. The RECON cost model was developed in Section 2.5 to compare constellations of equal cost and thus determine the best performance per unit cost. In doing so, the actual monetary value of each constellation was not critical to the results, since the costs were only compared relative to each other. It was acknowledged that the cost estimates were low, especially considering the historical costs of government satellite programs. The baseline cost calculated for an example 20 satellite constellation, able to provide 1 m resolution panchromatic imagery, was $768M.

A more realistic cost for this satellite constellation must be determined in order to perform a first order comparison with a UAV system. Based on historical data for U.S. Department of Defense (DOD) satellites, a satellite in the range of 400-600 kg may cost anywhere from $100M to $800M [6]. Unfortunately, cost information for the Earth-observing IKONOS and EROS missions is not readily available. Many ongoing DOD programs have experienced significant cost increases and delays. The Wide band Gap filler Satellite program will provide military communications; its total cost is near $2B for 5 satellites. Similarly, the Space Radar program, which now calls for 8 satellites, has a projected cost of $4B [12]. This equates to a per satellite cost of close to $500M.

The cost of a single high resolution Earth-observing satellite, for the purposes of this thesis, is taken to be $400M. A constellation of 20 satellites, assuming a learning curve in production (Section 2.5.4), costs approximately $4B.

3.4.2 Modern UAV System Capabilities

There are several UAV systems currently in operation by the U.S. Government. The MQ-1 Predator began as an Advanced Concept Technology Demonstration (ACTD) in 1994, and was further pursued by the Air Force for reconnaissance. The U.S. military used iterations of the initial Predator design extensively in Afghanistan and Iraq.

Teledyne Ryan Aeronautical began developing the RQ-4 Global Hawk in 1995 for a more focused surveillance role as a High Altitude Endurance (HAE) UAV [49]. The Global Hawk features Synthetic Aperture Radar (SAR), Electro-Optical (EO), and Infrared (IR) sensors and has a maximum flight time of 30.5 hours. The military has employed prototypes in operation, though there were noted technical failures. The government has given considerable attention to the cost and schedule overruns of the program, and the cost per aircraft, including development costs, now sits at $130.5M [48]. The Naval Research Advisory Committee quoted operations costs for the UAV as high as $26,500 per hour [1]. Despite these concerns, the program remains in development with models planned for the Air Force and Navy.

Among the currently existing UAV systems, the Global Hawk should serve as an appropriate vehicle to model the response times and performance of UAV systems. The sophisticated sensing hardware and HAE capabilities make it an appropriate reconnaissance system in the context of the mission.

Table 3.3: Global Hawk Characteristics [23, 48]

Parameter	Value
Flight Rate	0.179 km/s
Max Flight Time	28 hours
Max Range	22,000 km
Repair Time	6 hr
Operations Costs	26,500$/hr
Unit Cost	130.5 M$

3.4.3 UAV Architecture Analysis

This section discusses the performance characteristics of the Global Hawk UAV and introduce two designs for achieving the reconnaissance objective. A global distribution model is suggested for rapid response time, and a central storage and deployment model is also described as a lower cost option.

The intrinsic performance characteristics of a single UAV determine the responsiveness of a fleet of UAVs. Using the performance numbers from the Global Hawk UAV, as shown in Table 3.3, the responsiveness of a globally distributed fleet can be determined.

The maximum flight time is taken as several hours less than the demonstrated value of 30.5 hours, to account for any operational inefficiency. The repair time is approximated as simply a fraction of the maximum flight time, to allow for refueling and any maintenance. The reconnaissance mission requires responsiveness to any target on the Earth, and for the purposes of the UAV analysis this is set as the total land mass on the Earth (outside Antarctica), approximately 148.9 million km^2.

The mission duration - the duration of desired target coverage starting at initial notification - is set at 45 days. The goal of the UAV fleet is to arrive on target as quickly as possible, and then provide continuous coverage for the entire mission. The first vehicle is sent as soon as the target location is known. It provides coverage for as long as possible, until it must return to base to refuel. When it returns to base, it hands off the imaging responsibilities to a second UAV deployed to replace it. If there is sufficient time for the first vehicle to return to the base, refuel and be repaired, and return to the target location to replace the second, then only two UAVs are required. If not, continuous coverage may call for three UAVs.

A total program cost can be estimated by taking into consideration both the vehicle and operations costs, and using the appropriate number of vehicles to provide continuous coverage. This cost assumes that there will be two missions each year for ten years. This is the same assumption that is used in the RECON satellite analysis. These data are used to develop two models for UAV reconnaissance.

Global Distribution Model

One model for accomplishing this responsive reconnaissance task is distributing UAV systems evenly all over the globe so that in an emergency the closest station could respond. This may leave many units unused, but it allows for a very low response time. A simple relationship between the response time and the range to target is shown in Figure 3-3, based on the flight speed of the Global Hawk.

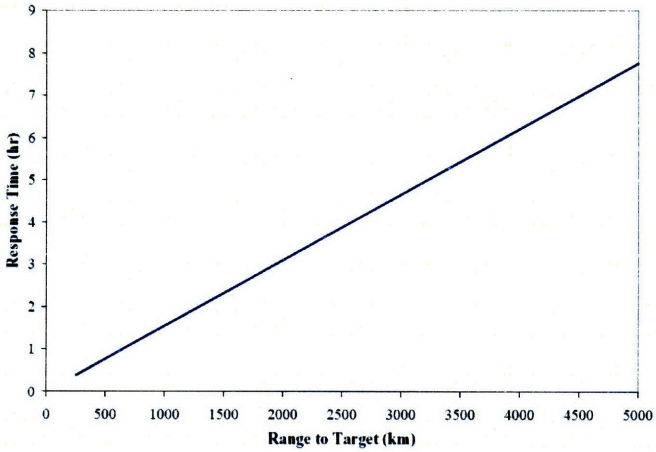


Figure 3-3: UAV Responsiveness to Target

The number of UAVs required for the mission is determined by first calculating the response area footprint for a single UAV. For a maximum range to target, r, this area is πr^2. If the land mass of the Earth is S, then the mission requires $S/\pi r^2$ UAV bases. The parameter r is the furthest distance that any target can be from a UAV base. The maximum range to target r determines the number of UAVs to be placed at each base. By varying the maximum range to target that each UAV base would be responsible for, the program cost changes dramatically, as shown in Figure 3-4. The program cost includes both vehicle purchase and operational costs, but does not include costs for installing new bases if necessary.

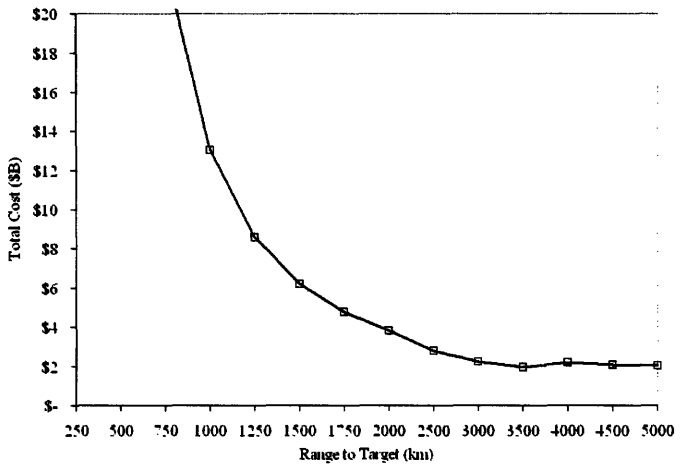


Figure 3-4: Total UAV Program Cost for Global Distribution

Note that there is a slight jump in cost near 4000 km range to target, because at this distance, three UAVs instead of two are required to provide continuous coverage. The minimum cost occurs at approximately 3500 km. At this design point, the maximum response time is near 5.5 hours, and each UAV operates for 17 hours over the target. Only eight UAVs are required in this case, and the total cost is $1.95B.

Note that this cost is based upon the Global Hawk, which currently has upwards of 50 vehicles planned for production. These high production volumes drive down the unit cost. Accordingly, if this reconnaissance mission requires only eight vehicles, then the $1.95B cost is only appropriate for the use of a standard or slightly modified UAVs already in production. If the new UAV were to be commissioned for this individual purpose, the costs would increase.

Shorter response times are possible if a larger UAV fleet is purchased. For instance, with a maximum range to target of 1000 km, the response time is 1.6 hours and total program cost near $13B. Because of the proximity of the base to the target, each UAV would be able to operate for 25 hours over the target. For this configuration, satisfying the maximum range to target over the Earth's land mass requires 95 UAVs.

Intrinsic in these calculations is the assumption of the ability to base several UAVs within 3500 km (or even 1000 km) of any target on the Earth's land mass. This is optimistic, and is constrained by physical base locations.

Central Storage and Deployment Model

A more cost-effective strategy is to purchase only three UAVs, and reposition them upon target notification. This solution sacrifices the initial response time in favor of having a smaller fleet of UAVs. The farthest relocation distance would be on the other side of the Earth, which requires a travel of 20,000 km. The maximum range of the Global Hawk is approximately 22,000 km. Assuming that a base exists in proximity to the target, the UAV fleet could make the initial trip in close to 30 hours under their own propulsion, or perhaps more likely via another transport mechanism. Thereafter, the UAVs can provide continuous coverage operating out of a local base. Three vehicles can accomplish this task for a total cost of approximately $500M.

3.4.4 Performance Comparison

A performance comparison is made between RECON and the two UAV architectures discussed in Section 3.4.3. Due to the additional capabilities of the satellite constellation - global coverage over time - over UAVs, a simple program cost comparison does not quite capture the situation. A more effective approach is to apply value to each configuration of the satellite system. In this way, the total cost can be considered a combination of the cost of the global coverage capability and the cost of the regional, or reconfigured, capability.

If it is assumed that half of the value of the RECON mission is in the ability to reconfigure, then a more direct comparison can be made to a UAV system. The performance characteristics of the satellite constellation and several UAV options are shown in Table 3.4. Qualitatively, the results can be succinctly summarized. In the case of RECON, a satellite might take the first image of the target within several hours, but the reconfiguration may not be complete for many days. Even after it is complete, coverage is not continuous. When using UAVs, the initial response times may be slower, but once on site the reconnaissance is continuous for the 45 day mission duration.

A simple cost comparison indicates that a low cost globally distributed UAV system fulfills the responsiveness mission requirements for the same cost as a satellite constellation. As expected, the response time of the UAV system is longer, but once in place the coverage is much higher.

The more responsive UAV option, meanwhile, seems excessive in terms of the relative

costs, while the centrally stored UAV option is not able to compete with the responsiveness of the satellite constellation. Considering the relative economy of the centrally stored UAV system, it may serve to complement the satellite system, but for reasons of response time it will not be able to replace it. Note that the importance of the response time depends on the nature of the event to be investigated; here it has been assumed that the emergency scenarios are extremely time-critical.

3.4.5 Political Issues of Aerial vs. Satellite Reconnaissance

Section 3.4.4 provided a discussion of the performance of both RECON and UAV systems. However, there are other elements to be considered when comparing the RECON and UAV options. Although the globally distributed UAV system cost is only half of the total satellite constellation cost and can fulfill the responsiveness objective, there are advantages to having a space based system.

The 1960 U-2 crisis, in which a spy plane was shot down over the Soviet Union, serves as a good example of the tensions which exist over airspace. Even very recently, in 2001, a U.S. Navy EP-3 surveillance aircraft was forced to make an emergency landing on Chinese soil after colliding with a fighter sent to intercept it. This incident garnered much publicity, and still today aerial reconnaissance remains a sensitive military strategy which has potential

Table 3.4: Satellite and UAV Performance Metrics

System	Config.	Descr.	# Units	Max. Response Time	Coverage per Day	Cost ($B)
RECON	N/A	N/A	20	First Image: 1.5 hrs, Reconfig. Complete: 15 days max	30 min. per day from 20 satellite passes	2.0*
UAV	Globally Distributed	Low Response Time	95	1.6 hr	Continuous	13.0
		Low Cost	8	5.5 hr	Continuous	2.0
	Centrally Stored & Deployed	N/A	3	30 hr	Continuous	0.5

*Assumes *half* the value of the RECON mission is in the ability to reconfigure and respond to events.

for creating further conflict.

Many countries very aggressively protect their airspace, and the ability of UAVs to perform reconnaissance missions, even with stealth technology, remains in question. Using aircraft for reconnaissance in military situations remains a risky endeavor, and depending on the political sensitivity of the situation, satellite imagery may be the only reliable way to gather intelligence. A 2003 U.S. government report was very clear with this message, stating that "U.S. national security is critically and increasingly dependent upon space systems," and additionally that "UAV and other programs complement, but do not replace space [18]."

3.4.6 Summary

Many of the conclusions regarding this comparison depend heavily on the value that is placed on responsiveness. A globally distributed system of UAVs may be able to replace a satellite constellation's ability to reconfigure by sacrificing response time for continuous coverage. A system of eight vehicles can be implemented at a reasonable cost and provide a response time of 5.5 hours. Central storage and deployment of UAVs offers an even more economical solution with a three vehicle system that can be implemented for a fraction of the RECON mission cost. This setup would provide a maximum response time of 30 hours, and complement the capabilities of a satellite system. Issues of airspace, however, will likely inhibit many of the desired reconnaissance missions if UAV stealth technology is unavailable. In military scenarios where political tensions are too high to risk the uncovering of a spy UAV, a satellite system can be deployed more discreetly. Furthermore, if both space and aerial systems were selected to work in concert, the military may opt to design down the RECON system to accommodate fewer reconfigurations. Situations which are not time-critical and where the target is available for over-flight might call only for UAV deployment.

Given the limits of UAV reconnaissance, especially for military applications, it may be best to consider the two methods as complementary and not look to choose one over the other. With this approach, the question then becomes whether the performance improvements generated by reconfiguration are worth the extra expense when only considering the utility of the satellite imagery. This question stands regardless of what any UAV system, because of its politically-based unreliability, might be able to provide.

Chapter 4

Image Collection Planning

Interest has grown recently in developing lightweight, modular space telescopes which are more cost-effective than traditional monolithic telescope designs. As space systems become less massive, however, they also become more flexible. The added flexibility affects the maneuverability of the satellite, as longer times are needed to allow vibrations to settle. This is especially a concern for low altitude Earth-observing satellites, which maneuver frequently and have limited access times to desired targets. The optimization of imaging for these satellites is subject to orbital, slewing, and geometric constraints.

The objective of this chapter is to develop accurate models of the image collection optimization problem - also called the satellite scheduling problem. Graph search, dynamic programming, and linear programming techniques are used to model the optimization problem. Algorithm modifications are additionally proposed to model the use of a steering mirror to enhance the satellite's field of regard. Implemented optimization methods are employed in Chapter 5 to study the implications of agility and flexibility in the design of a lightweight space telescope.

4.1 Introduction

Satellite systems are a very expensive resource, and it is important that they be operated efficiently. Mission planning and scheduling is critical to the success of any space mission. For astronomical telescopes, observations may occur over very long time durations to detect light from distant stars. In this case, the time spent imaging is a large fraction of the total time, while the time spent slewing to observe new targets comprises a relatively smaller

fraction. When considering terrestrial science targets, this may not necessarily be the case. Imaging times are shorter, and the agility of the satellite plays a much larger role in the number of images that can be collected during a given time period. It is therefore important to consider the effects of different design parameters on the agility of a space telescope, which in turn determines the overall imaging efficiency.

Previous work has been done on the design of lightweight space systems. The Modular Optical Space Telescope (MOST) is a tool which enables rapid trade-space exploration for the design and analysis of a lightweight space telescope [61, 15]. Both monolithic and segmented aperture configurations are considered, and the model is parameterized by variables such as the focal length and mirror areal density, among many others. MOST uses an integrated approach, developing a Finite Element Model (FEM) for each telescope instantiation, running NASTRAN to extract the satellite normal modes, and performing analysis to calculate performance metrics. Various control algorithms are used to compensate for the flexibility of these lightweight systems. Still, the reconfiguration time for the satellite - the time it takes to slew to a new target and allow vibrations to settle - is a strong function of the satellite design.

In order to first address the interaction between satellite agility and imaging performance, several optimization formulations are developed using graph-search, linear programming, and dynamic programming techniques. Each determines the performance of a given satellite design in imaging a set of targets. The advantages and disadvantages of each approach are discussed in terms of modeling accuracy and computational performance.

The question that Chapters 4 and 5 aim to answer are:

- How does the design of the satellite affect its agility?
- How do the changes in satellite agility affect operational performance?
- What design heuristics can we apply in future analysis?

To answer these questions, the following approach is taken: 1) develop tools to analyze operational satellite performance for appropriate scenarios; 2) use the MOST model to explore different design parameters; and 3) analyze the operational performance of these designs.

Section 4.2 reviews previous work that has been done in satellite scheduling (image collection optimization). Section 4.3 defines the problem accurately and Section 4.4 presents

three approaches for solving it. A comparison of these approaches is presented in Section 4.4.4. Section 4.5 describes extensions which allow the modeling of a fast steering mirror (FSM) as part of the optimization framework. Preliminary analyses using the FSM are presented to demonstrate potential operational performance improvements.

Chapter 5 builds upon the work presented in this chapter to compare the imaging productivity of different satellite designs.

4.2 Literature Review

A major operational task for any satellite is to schedule events (e.g. image target i, down link to ground station, calibrate star trackers) such that mission requirements are met and optimal use is made of resources. This problem falls into the more general category of resource allocation. An example problem from a different domain is the scheduling of a professional sports league season. The resources (stadiums) must be assigned to pairs of teams such that the correct number of games are scheduled, and each team is able to play every other team. Resource allocation problems can be solved by a variety of tools in optimization, including linear programming, dynamic programming, and constraint satisfaction.

There is extensive literature on image collection optimization in space systems. Partial enumeration of feasible schedules is a simple approach that is satisfactory for small problems [29]. Graph search algorithms have been used to model the satellite scheduling problem and also find best imaging paths based on multiple criteria [20, 22]. Efforts have been made recently to focus on the difficulties introduced by so called agile satellites, which are able to maneuver in three dimensions. Satellites such as IKONOS and the recently launched WorldView-1 take advantage of this and are able to image significantly more targets of interest. This additional flexibility comes at the cost of added scheduling complexity. Linear programming techniques have been suggested [21, 19, 43]. Integer programming with Lagrangian relaxation has been used to model the problem with additional criteria including storage and power constraints [42]. Constraint Satisfaction (Constraint Programming) has also been investigated [51]. Verfaillie and Lematre consider several approaches, including greedy, dynamic programming, and constraint programming algorithms to tackle this problem [39]. Local search techniques, such as Tabu search and genetic algorithms, have

also been investigated [28, 68, 62, 41, 27, 40]. Hybrid algorithms have been pursued to combine the best aspects of two different approaches to improve the solutions [42]. Some work has been done to compare many different approaches [7, 26, 63, 24]. Martin provides an overview of the image collection optimization process as it applies to the operational IKONOS satellite [44].

The process for selecting specific targets to image may take place over the course of a full month. The initial target set may contain upwards of 20,000 locations [44]. Monthly and weekly refinements are made to the schedule. Finally, a daily schedule is produced that identifies the sequence of events for the day, allotting time to imaging windows while making sure to satisfy thermal, power, and capacity constraints. The focus in this chapter is on the optimization of imaging events within a single imaging window. It is assumed that potential imaging targets are randomly distributed around the Earth's landmass. As the objective is to perform rapid analysis and comparison of satellite designs, additional constraints such as satellite thermal limits and data storage capacity are not considered here. In theory, a full satellite scheduling system with all physical constraints could be applied to evaluate various designs, but computation times would be prohibitive.

It should be noted that it is reasonable to consider both perfect and approximate optimization approaches. Stochastic algorithms (e.g. Simulated Annealing) yield near optimal solutions in significantly less runtime than their optimal counterparts. Due to the randomized and non-optimal nature of these algorithms, however, it is difficult to make conclusions about the performance of a system based on a small set of scenarios. A Monte Carlo method must be used. The computational performance savings gained by approximating the solution are lost in this way. The approach taken in this thesis is to use optimal problem formulations. Several methods are presented in an effort to 1) compare runtime performance of different implementations and 2) provide validation of successively sophisticated models.

4.3 Problem Definition

The general image collection optimization problem is presented. This problem forms the basis for this chapter, and is investigated with several different algorithms in Section 4.4.

Consider a satellite at an altitude h. The satellite field of regard is defined by a target

minimum elevation angle, η. The satellite must be above the target local horizon by an angle of at least η in order for an image to be captured . The field of view, FOV - the size of the instantaneous sensor footprint on the ground - is defined by the on board optics. As the satellite orbits the Earth, it sweeps out an area proportional to its orbital velocity, v, and field of regard. This is illustrated in Figure 4-1. During this time, it has an opportunity to image $i = 1...N$ targets, each at a different geographic location. Each target has an associated priority w_i. The objective is to maximize accumulated w_i during the given time window, subject to constraints on geometry and satellite agility. Targets 1 and N are considered dummy targets which indicate the starting and ending position of the satellite. Figure 4-2 shows the geometry of the problem. The following are relevant problem variables.

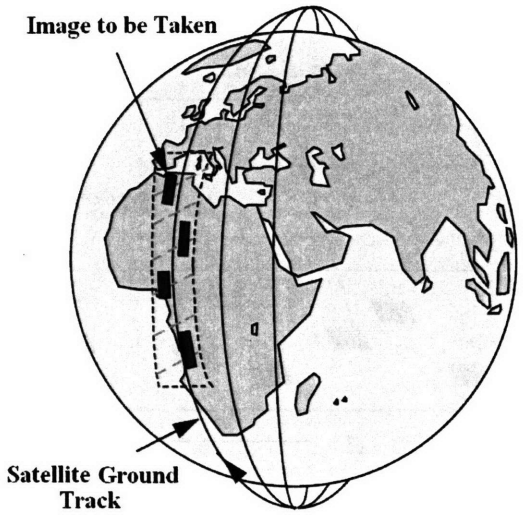


Figure 4-1: Imaging Window of an Orbiting Satellite [22]

- N: Number of targets to consider.
- x_i, y_i: The location in the ground plane of target i.
- t_i: The time at which target i is imaged.
- a, b: The dimensions of image-able area.
- d_i: The imaging time required for target i.
- s_{ij}: The transition time required between imaging targets i and j.
- w_i: The priority of imaging target i. Taken as an integer, $1 < w_i < 10$.

- η: The minimum target elevation angle.
- v: Satellite orbital velocity.
- h: Satellite altitude.

This problem has several important aspects. It is a combinatorial problem since finite combinations of images can be selected. It is also a non-linear problem because of the complicated satellite dynamics between images taken at different geometries. Section 4.4 pursues three methods for solving the image collection problem. A graph search and integer programming algorithm are developed under some simplifying assumptions before considering a more general problem using dynamic programming.

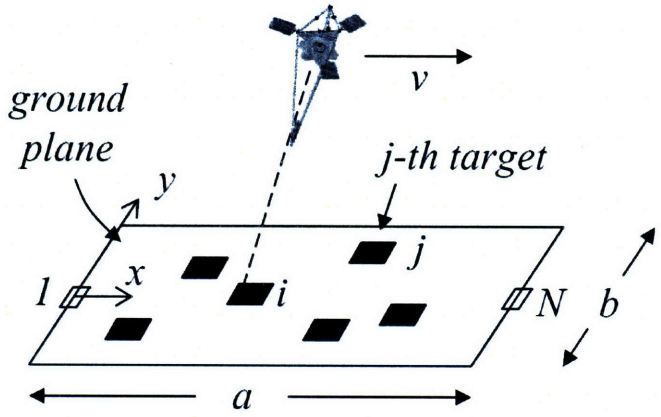


Figure 4-2: Geometry of the Image Collection Optimization Problem

4.4 Implemented Algorithms

There are many different approaches which can be taken to solve the problem described in Section 4.3. This section describes three algorithms in the interest of identifying the merits of each approach and common assumptions that are made. The performance of each is investigated in Section 4.4.4.

4.4.1 Depth-First Branch and Bound

This section describes the formulation and implementation of a graph search algorithm to solve the Image Collection Optimization problem.

One set of algorithms that can be applied to this problem are graph search algorithms. Graph search is fundamental to computer science and artificial intelligence disciplines. The abstract definition of a graph search problem is as follows [54]: consider a graph which consists of a set of vertices V $(i = 1 \ldots N)$ connected in some fashion by a set of arcs $(i, j) \in E$. The objective is to find a path between the start node S and the goal node G.

In order to explore the graph in an attempt to find a feasible path, a search algorithm often begins by expanding all paths that can leave S. In some prescribed order, the algorithm expands all of these new paths, at each step checking to see if G has been reached. Common search orderings are Breadth-first and Depth-first. Breadth-first search systematically explores paths nearest to S, expanding outwards. Depth-first, on the other hand, aggressively expands outward from S. Depending on the structure of the tree, one algorithm may perform significantly better than another. In general, Breadth-first search has higher memory requirements, while Depth-first requires more computation time.

To apply this structure to the image collection planning problem, the assumption is made that the satellite is constrained to image only by rotating in the roll direction, around its velocity vector. That is, images are only taken exactly when the satellite passes the target, in azimuth of either $\pi/2$ or $-\pi/2$. Figure 4-3 illustrates the roll and pitch angles for an orbiting satellite. It will be explained later how this assumption can be relaxed. Each image is taken with the roll-rate of the satellite equal to zero. Consider the geometry shown in Figure 4-2. In this figure, the satellite enters the imaging area at $x = 0$, with the velocity vector in the positive x direction. There are $N - 2$ targets distributed over the total area. These targets are labeled in ascending order from left to right, such that $x_j \geq x_i$.

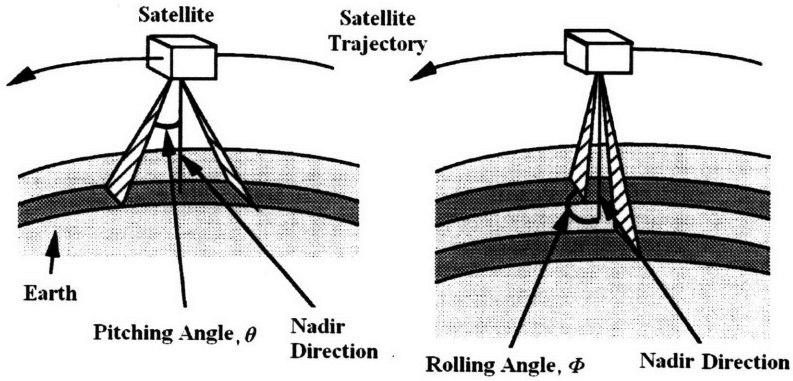


Figure 4-3: Satellite Attitude Definitions [20]

Given characteristics of the satellite agility, the feasibility of all imaging transitions is determined. The required transition time s_{ij} is calculated based on the angular change that the satellite must undergo between targets i and j, as well as the satellite agility. A mathematical description of the graph is given in Equations (4.1).

$$1 \ldots N \in V \tag{4.1a}$$

$$\phi_{ij} = \left| \arctan \frac{y_j}{h} - \arctan \frac{y_i}{h} \right| \tag{4.1b}$$

$$s_{ij} = f(\phi_{ij}) \tag{4.1c}$$

$$s_{ij} \leq \frac{x_j - x_i}{v} \implies (i,j) \in E \tag{4.1d}$$

$$(1,j) \in E \quad \forall j = 2 \ldots N \tag{4.1e}$$

$$(i,N) \in E \quad \forall i = 1 \ldots N-1 \tag{4.1f}$$

The value ϕ_{ij} is the angle that the satellite must traverse in order to image targets i and j in sequence. The function f characterizes the agility of the satellite. It specifies the time required, s_{ij}, for the satellite to perform a maneuver of magnitude ϕ_{ij}. If s_{ij} is less than the time it takes for the satellite to travel from x_i to x_j ($\frac{x_j - x_i}{v}$), then the transition is feasible, and the arc (i,j) is thus a member of the feasible transition arc set E. We assume imaging time, d_i, to be negligible. All transitions exiting node 1 and entering node N are automatically feasible, since 1 and N are dummy targets which represent the satellite entering the imaging area. The start node S is defined as vertex 1 at the left edge of the ground plane in Figure 4-2. The goal node G is vertex N at the right edge of the ground plane. At this point the graph structure has been defined, and the outcome of any graph search algorithm applied to this tree is a traversable path from node 1 to N.

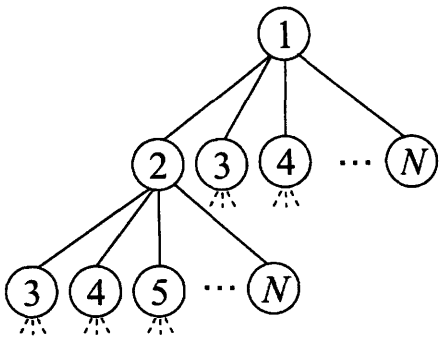


Figure 4-4: Depiction of the Graph Search Problem

The general structure of the search tree is shown in Figure 4-4. Specific branches are pruned based on the feasible transitions in E. The specific search method implemented here is Depth-first. Appendix B contains details regarding this choice. Nodes with lower values of i are expanded first, as this generally results in a path that images more targets. In order to find an optimal path, however, we must consider also the target priorities, w_i. Instead of terminating upon reaching the goal node G, the algorithm instead stores the path and corresponding value, and continue to explore the search tree. At each successive discovery of the goal node G through a unique path, the algorithm compares the current path to the currently known optimal path. If the current path is of higher value, the stored optimal path is updated. This continues until all paths have been explored.

This exhaustive search is very computationally intensive. To improve the search, consider first how the graph is explored. As the search algorithm explores paths extending from S, a set of new partial paths is added to the search queue for later exploration. Each partial path from S to an intermediate node i has an associated partial cost - the cost accumulated by taking the given path from S to i. When the path to node i is extended to node j, the total accumulated cost is updated.

In order to accelerate the search, a branch-and-bound method is employed to eliminate inferior paths. The information regarding the cost of partial paths to intermediate nodes can be used to improve the search efficiency. A cost vector c of length N is initialized to zero. The value $c(i)$ represents the score of the best possible path to node i that has been discovered thus far. As paths are extended from S, the cost of each partial path is stored. Each time a path is expanded to a new node j, the partial cost of that path is compared to $c(j)$. If $c(j)$ is greater, then a better path from 1 to j has already been found, and the current path is discarded. If $c(j)$ is less than the value of the partial path in question, then $c(j)$ is updated accordingly and the partial path will continue to be explored.

The graph search method is simple to implement. Appendix E contains source code for an implementation in MATLAB®. Additional modifications can be made to allow for slewing in the satellite pitch direction as well (Figure 4-3). Consider a graph with a vertex set V described by two pieces of information: the target index i and an imaging time k. In a similar fashion, a graph can be set up such that $(i,k), (j,m) \in E$ where the transition time $s_{(i,j),(k,m)}$ between (i,k) and (j,m) is less than m - k. This idea is further explored and implemented using dynamic programming in Section 4.4.3.

4.4.2 Integer Programming

Another common approach in optimization problems is linear programming [9]. Linear programming is a very extensive field, and even within the field of satellite scheduling there have been many approaches provided. Verfaillie and Lematre suggest a method based on a modified Traveling Salesman Problem [39]. In general, the way that the optimization problem is defined - the selection of decision variables and design of constraints - can have a significant effect on the performance.

Consider the same problem as discussed previously. The satellite is constrained to image only in the roll direction, and must optimally choose from a set of N targets. One way to phrase this problem as a linear program is shown in Equation (4.2). Define z_i as a binary variable evaluated as one if target i is imaged.

$$\max \sum_{i=1}^{N} z_i w_i \tag{4.2a}$$

$$z_i + z_j \leq 1 \quad \forall (i,j) \notin E \tag{4.2b}$$

$$z_i \in \{0,1\} \tag{4.2c}$$

The main constraint (4.2b) on the maximization problem enforces the fact that if two targets i and j cannot be imaged consecutively, then at most only one of them can be imaged. Although this formulation is valid, there exists a way of describing the optimization problem that in practice performs better. This is the result of differences in the shape of the feasible optimization space, in particular the way that solvers use branch and bound to find a solution [9].

Instead of identifying which targets are imaged as a single variable z_i, the new formulation uses a binary variable x_{ij} to represent if target j is imaged after target i. If $x_{ij} = 1$, then both i and j are imaged, in that order. Note that this definition implies that the objective is to find a route through the targets from 1 to N. This is indeed analogous to the Traveling Salesman Problem (TSP). In the standard TSP, the goal is to find the shortest path that visits all N points in a plane exactly once, and returns to the starting point. As in the TSP, constraints that enforce the concept of a feasible path through a set of points are implemented. The main differences are that in the satellite scheduling problem, it is possible that only a subset of the points may be visited. Furthermore, the objective is to

find the longest (or highest value) path. Equation (4.3) is adapted from [39] and describes the objective function and constraints on the problem.

$$\max \sum_{i=1}^{N} x_{ij} w_i \tag{4.3a}$$

$$\sum_{i=1} x_{ij} - \sum_{k=1} x_{jk} = 0 \; \forall \; j = 1 \ldots N \tag{4.3b}$$

$$\sum_{i=1} x_{ij} \leq 1 \; \forall \; j = 1 \ldots N \tag{4.3c}$$

$$\sum_{j=1} x_{1j} = 1 \tag{4.3d}$$

$$\sum_{i=1} x_{iN} = 1 \tag{4.3e}$$

$$x_{ij} = 0 \quad \forall \; (i,j) \in \{j > i\} \tag{4.3f}$$

$$x_{ij} = 0 \quad \forall \; (i,j) \in E \tag{4.3g}$$

$$x_{ij} \in \{0,1\} \tag{4.3h}$$

Constraint (4.3b) enforces the concept of a feasible path on the solution to the problem by first stipulating that the number of paths into node j must equal the number of paths exiting node j. Constraint (4.3c) supplements this by ensuring that there is at most one path into any node. Combined, the first two constraints result in a solution with one path. Constraints (4.3d) and (4.3e) are the boundary conditions. There must be a path leaving node 1, and there must also be a path entering node N. Constraint (4.3f) ensures that target j can be imaged after target i only if $i < j$ and constraint (4.3g) ensures that the agility constraints are met. In implementation, constraints (4.3f) and (4.3g) may be enforced by simply eliminating all relevant decision variables x_{ij} for $(i,j) \notin E$ prior to solving the optimization problem.

4.4.3 Dynamic Programming

Up until this point, this section has considered the limited case of imaging only in the roll direction. Recently launched satellites are able to acquire images in both the pitch and roll directions. This capability enables more efficient collection sequences, but also significantly complicates the optimization space. Dynamic programming is another tool that can be applied to solve these more complex optimization problems [8]. Although

dynamic programming may be used on the restricted scenario, here it is implemented to support the more general case.

The basic elements of dynamic programming are the idea of *states* and *actions*. Each state has a corresponding set of actions, which may be null. Each action defines a legal transition to a new state. Additionally, each action has a prescribed cost. The problem is to find an optimal set of actions to carry the system from an initial state to a final state. In the satellite scheduling problem, we define a state as a combination of the target to be imaged, i, and the time when the image is taken, t_k. The state is a pair (target, time) $= (i, t_k)$. The target comes from the set $i = 1 \ldots N$. The time may be any discrete value within the imaging window of the target. The problem as formulated by dynamic programming is described in Figure 4-5.

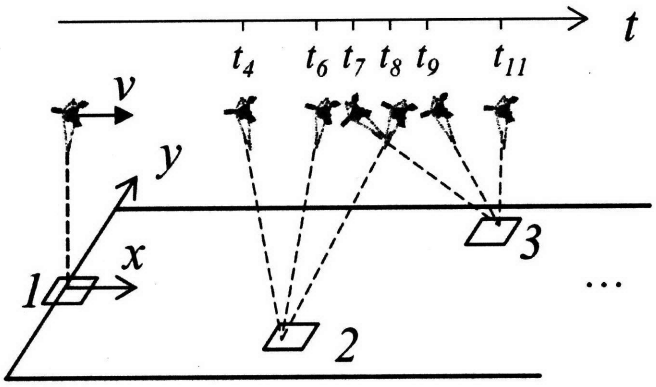


Figure 4-5: Definition of States Under Dynamic Programming

Feasible state transitions are determined based on the geometry of the problem, similar to the method described by Equation (4.1). For any two pairs of states (i, t_k) and (j, t_m), the transition is feasible if Equation (4.4) holds.

$$d_i + f\left(\phi_{ij}\left(t_m, t_k\right)\right) < t_m - t_k \implies \left(\left(i, t_k\right), \left(j, t_m\right)\right) \in E \tag{4.4}$$

Here, d_i represents the imaging time required for target i, and is nominally set to zero. The function ϕ_{ij} represents the satellite attitude change from imaging target i at t_k to target j

at t_m. This is based on the geometry between the two targets, and is calculated as:

$$\begin{aligned} \phi_{ij} &= cos^{-1}\left(\frac{r_i \cdot r_j}{|r_i|\,|r_j|}\right) \\ r_i &= \left[v(t_k - t_i') \;\; y_i \;\; -h\right] \\ r_j &= \left[v(t_m - t_j') \;\; y_j \;\; -h\right] \end{aligned} \tag{4.5}$$

where t_i' and t_j' are the times at which the satellite passes targets i and j. In order for the transition to be feasible, the required time must be less than the available time. This is similar to the approach taken in [39], though here the transition times between targets are not assumed constant. Making the assumption of constant transition time eliminates the need for computation of ϕ_{ij} for all state pairs, but reduces the accuracy of the formulation.

All transitions leaving the dummy initial state $(1, t_1')$ as well as all entering the dummy final state (N, t_N') are set as feasible. The function f characterizes the agility of the satellite as described in Section 4.4.1. The value for transitioning to state (j, t_m) is simply w_j, the value of imaging target j.

An example graph constructed by this process, corresponding to Figure 4-5, is shown in Figure 4-6. Each box represents a (target, time) pair, and the lines represent feasible state transitions. The parameters w_j correspond to the value of the transition. Note that the transition (2,8) to (3,7) is infeasible regardless of the satellite agility. This is because it is impossible to image target $i = 2$ at time t_8 and *then* target $j = 3$ at the earlier time of t_7. In the example graph, (2,4) to (3,7) does not meet the dynamic constraint in Equation (4.4). There is insufficient time to reposition the satellite from imaging target $i = 2$ at time t_4 to imaging target $j = 3$ at time t_7.

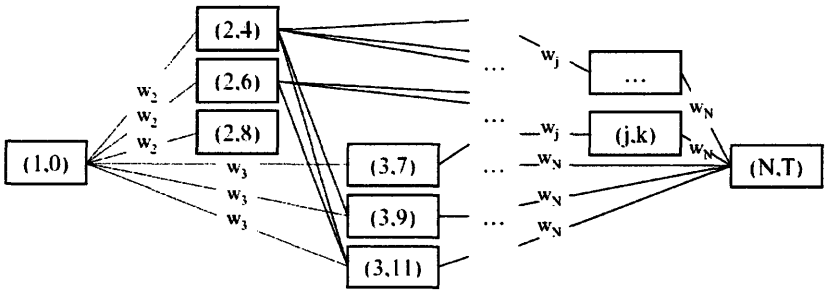


Figure 4-6: Example Graph Used for Dynamic Programming

In general, the reward for capturing an image is set based on the geometry of the

satellite-target pair at time t_m. For example, the image j is given a maximum value of w_j when the geometry of the satellite and target provide maximum resolution (the target is at satellite nadir during imaging). The value tapers off to $0.5w_j$ at the edge of the field of regard. Note the special case when the imaging time is restricted to take only one value for each target, corresponding to the time when the satellite passes the target ($t_k = t'_i = x_i/v$). This corresponds to imaging only in the roll direction, and the problem simplifies to the roll-restricted case.

To ensure that the path meets the end constraints, a value of $10 * \max_i(w_i)$ is assigned to any transition that reaches node N. A shortest path algorithm that iteratively converges to the optimal solution is applied to this problem [8]. The general form is given by Equation (4.6).

$$\begin{aligned} V(s)^0 &= 0 \ \forall s \\ V(s)^{k+1} &= \max_{a_i} \left[r(s, a_i) + \gamma V(s')^k \right] \end{aligned} \tag{4.6}$$

For each iteration and for each state s, a maximum is taken over all possible actions a_i, where a_i is a transition from state s to state s'. The value maximized is the reward $r(s, a_i)$ for taking action a_i plus the discounted current value of the state s'. A tolerance ϵ is set for the termination criterion. The algorithm terminates when $\left\| V^{k+1} - V^k \right\| < \epsilon$. The optimal actions a_i are stored for each state s during execution, and the algorithm output is the optimal action to be taken at any state. It is clear that the most value is attained by starting at state (1,0), earliest in the sequence.

The process for finding the optimal solution using dynamic programming is outlined below:

1. Determine imaging interval for each target, based on field of regard.
2. Discretize time over the entire interval, disregarding points where no target is in view.
3. For each target-time pair, determine all feasible transitions to other pairs.
4. Set the appropriate reward for each transition.
5. Apply the dynamic programming (shortest path) algorithm to the constructed graph.
6. Extract the set of actions from the solution.

Appendix E contains a MATLAB® implementation of this dynamic programming algorithm.

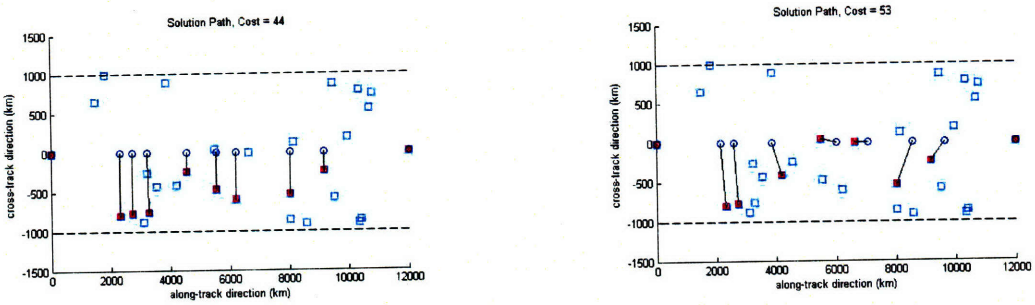
It is important to note that information is lost by discretizing time over the interval. The global optimal path may require a set of actions to occur at extremely precise times, which are lost when discretizing. Nevertheless, this is a common approach taken for the sake of good computational performance. Care must be taken to identify the proper time step resolution which balances these two aspects.

4.4.4 Performance and Validation

This section describes the performance of the problem formulations introduced in Sections 4.4.1 through 4.4.3, demonstrates successive validation of each, and recommends the best algorithm for use in trade-space analysis. Several hundred trials were run to ensure that all three formulations return the same optimal path. Note that if two paths exist with the same optimal value, the depth first branch and bound method has the capacity to return both, while the others do not.

Figures 4-7(a) and 4-7(b) show the optimal paths plotted for a roll-restricted case and a roll-relaxed case on the same set of targets. Target priorities w_i are selected as a random integer between 1 and 10. The dimensions a and b are set to 12,000 km and 2,000 km, respectively. The minimum elevation angle is set as $\eta = 30°$, such that images near the fringe of the area are accessible. The darkened boxes indicate images taken. The circles represent the satellite position at the image time and the lines connect satellite-target pairs. As shown, the value of the roll-relaxed optimal path (cost = 53) is a 20% improvement over the roll-constrained case (cost = 44). The optimal roll-relaxed path corresponds to imaging seven targets in 30 minutes, or one image every 4.3 minutes. As a point of comparison, the IKONOS satellite captures approximately 600 images per day [55]. Assuming that only half of the day is spent with access to valued targets (when the satellite is not over water or the poles), then the imaging rate is approximately once every 1.2 minutes.

Figure 4-8 shows the average run-time performance of each algorithm in finding an optimal solution for N targets, as N increases. The analysis was run in MATLAB® on a Pentium 4 3.0 GHz computer with 1.0 GB of RAM. The performance data for dynamic programming is based on the same roll-restricted case as described for graph search and integer programming. Both integer programming formulations grow very rapidly with an



(a) Roll-Restricted Case

(b) Roll-Relaxed Case

Figure 4-7: Optimal Paths for $N = 30$ Targets

increasing number of targets. This is to be expected, given that the Traveling Salesman Problem on which they are based is N-P hard. Depth-first branch and bound graph search performs better than both integer programming formulations, while dynamic programming performs far and away the best. Because the structure of each solution method remains the same as the complexity increases (e.g. imaging in the pitch direction), the performance comparison translates accurately to more general roll-relaxed implementations.

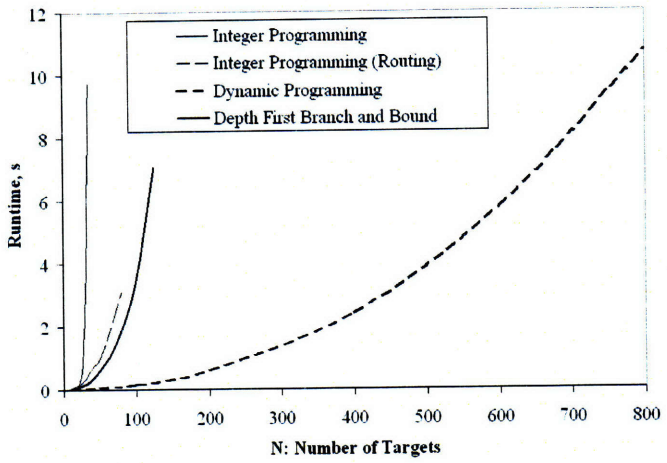


Figure 4-8: Runtime Performance of Implemented Algorithms for Roll-Restricted Case

Extending the capability of dynamic programming by relaxing the roll-constraint increases the number of states in the problem, and thus the run time. Because time is discretized, sufficient time resolution must be used to produce optimal paths. Figure 4-9 shows an example, for the same set of targets, of path value with increasing time resolution (fewer discrete time steps). Smaller time steps expand the number of unique states, and thus can increase optimal path value. Even after providing sufficient time resolution in the

roll-relaxed dynamic programming problem, the performance remains comparable to the graph search and linear programming roll-restricted methods.

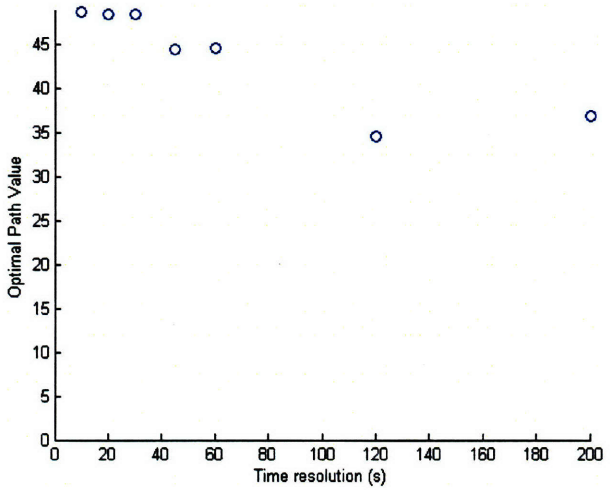


Figure 4-9: Decrease in Optimal Path Value with Increasing Time Steps

It is concluded that dynamic programming is a suitable method to pursue in comparing the performance of Earth-observing satellite designs. Chapter 5 introduces an approach for analyzing the imaging productivity of different satellite designs, based on their slew and settle properties.

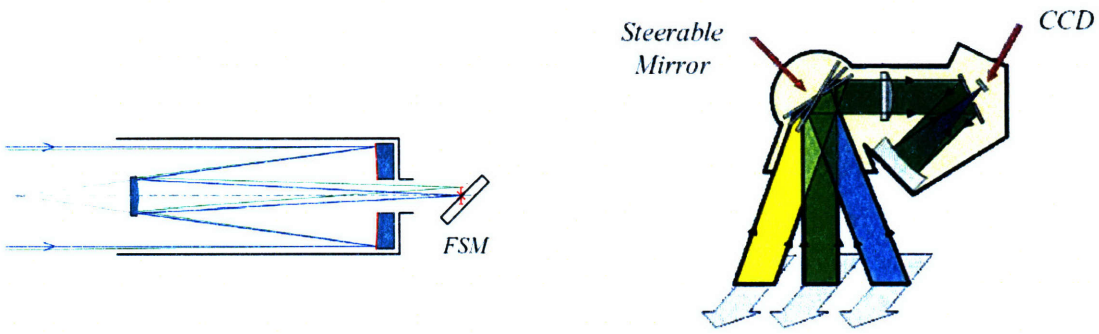
4.5 Optimization Extensions to Include a Steering Mirror

One additional component that may be used in an advanced optical system is a fast steering mirror (FSM). The main purpose of a FSM is to remove tip/tilt wave-front error from the image [34, 13]. This error is the result of oscillations of the telescope bore-sight around the desired imaging angle. In addition to minimizing tip-tilt error, a secondary function could be to extend the field of regard of the optics without slewing the satellite. Due to the small reaction mass, and the possibility of providing re-actuation, slewing a FSM incurs substantially less excitation of flexible dynamics and requires much less power.

There are two constraints on the usable range of the FSM. One is the physical hardware range constraint on the FSM. The second, and more likely binding constraint, is the result of the optics design. There is a fundamental limit on the angle at which off-axis light can be resolved. The limiting constraint is that the off-axis light must be focused on the CCD

detector. Appendix C contains a description of these constraints for a Cassegrain mirror design. For a system with an f-number of 2.0, the mirror can focus light up to $\theta_{l,max} = 2.9°$ off-axis. This value of $\theta_{l,max}$ translates, for a 500 km orbit, to approximately +/-25 km range of motion on the ground.

A more aggressive approach to extending the capability of the satellite is to place a steering mirror in front of the optics. The French SPOT satellites uses this technique, although limited to one rotational axis. SPOT collects imagery by rotating a steering mirror located in front of its optics, while the satellite remains pointed at nadir [58]. One can imagine a scenario, as discussed earlier, where both the satellite and steering mirror are able to rotate in two dimensions. Figure 4-10 illustrates two possible locations for the steering mirror.



(a) Behind Optics Near the Focal Plane

(b) In Front of the Optics [58]

Figure 4-10: Possible Locations for the Steering Mirror

Imaging with a steering mirror has the potential to induce optical errors, for example coma errors in the case of small mirrors located behind the primary. Nevertheless, in many cases it is preferred to use the steering mirror, despite this penalty, instead of slewing the satellite. A FSM requires much less power than reaction wheels. Additionally, since the mass and inertia of the FSM is small, slewing the FSM does not excite vibrations in the satellite. The full satellite is a high inertia object that carries a large penalty in both time and power consumption for slewing.

The objective in this section is to develop optimization frameworks to consider this problem. Two approaches are proposed, and the advantages and disadvantages of each are discussed. Analysis is done to determine the impacts of a FSM on overall imaging productivity.

4.5.1 Dynamic Programming

The dynamic programming formulation discussed in Section 4.4.3 is easily extended to consider a FSM. The state is defined as a triplet (target, time, fsmpos). The third element of the state indicates the two-axis orientation of the FSM, discretized into a user-defined number of angles. Each FSM position corresponds to a precise satellite attitude necessary to properly align the target. Note that fsmpos is interchangeable with the satellite attitude. Either can be used to define the third state element. Figure 4-11 describes the imaging options for a single target with fsmpos discretized as four values, where $\theta_l < \theta_{l,max}$. As previously, the state transitions are calculated using a definition of the satellite agility. For each triplet, all feasible transitions to other triplets are calculated.

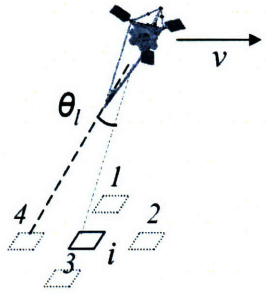


Figure 4-11: FSM Pointing Options for Target i

The reward function can be designed to account for the specifics of each state (FSM position). In the analysis to be shown, images taken with the FSM in a position other than along the telescope bore sight are penalized slightly. This is done to account for coma optical errors induced when using small steering mirrors located behind the primary mirror. In the case of large steering mirrors in front of the optics, there may also be particular imaging geometries to avoid, and excitation of flexible dynamics is much more of a concern.

The problem is solved using a shortest-path algorithm, as given by Equation (4.6). A trade off must be made between the resolution of the fsmpos variable, the time discretization, and the number of targets in the problem. Additional discretization is needed as the FSM range increases in order to sufficiently capture all of the feasible imaging geometries. If the number of FSM angles is set to zero, the problem reverts to the one discussed in Section 4.4.3.

As in the dynamic programming formulation described earlier, one of the shortcomings of this problem is due to the discretization of states. Figure 4-12 shows two targets i and j

which have the potential to be imaged sequentially.

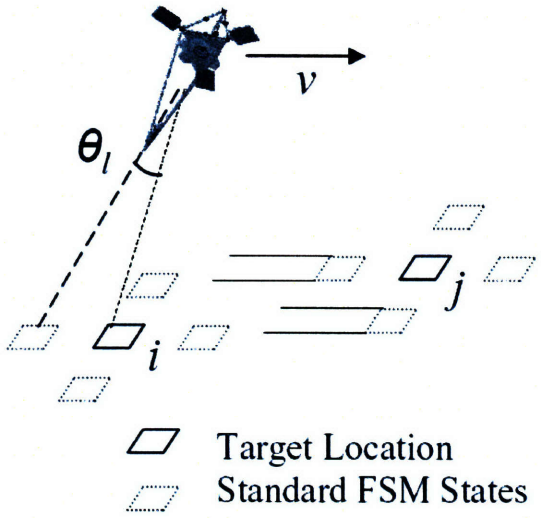


Figure 4-12: FSM Pointing Options for Targets i and j

There are many combinations of state pairs (i, t_k, fsm_1),(j, t_m, fsm_2) to be considered between these two targets, subject to the agility constraints. All require some slewing of the satellite, if the FSM is not available, since the states do not align in the direction v. Because even small motions require significant time to execute, a preferred option would be to only move the FSM to acquire targets i and j. This aspect may be incorporated by propagating the appropriate states downstream of target i such that there are imaging options for target j which do not require satellite motion. Figure 4-13 gives an example for two targets. The default imaging states prescribed for targets j_1 and j_2 each require slewing of the satellite. It may be the case that all of the transitions between i and j_1 are infeasible since there is insufficient time to slew the satellite. In order to take advantage of the benefits of the FSM, the imaging states corresponding to target i at fsm_0 and fsm_3 are propagated toward target j_1. New states with fsm_5 and fsm_6 are added such that the transition between states (i, t_k, fsm_0) and (j_1, t_m, fsm_5) (as well as between (i, t_k, fsm_3) and (j_1, t_m, fsm_6)) are automatically feasible, because no slewing of the satellite is required. The FSM, by virtue of its design, is slewed nearly instantaneously to image both i and j_1.

For situations where i and j are in close proximity, imaging both may only be possible through use of the FSM. It is important, then, that this possibility is captured. For situations where targets i and j are separated by a significant distance, this option, though

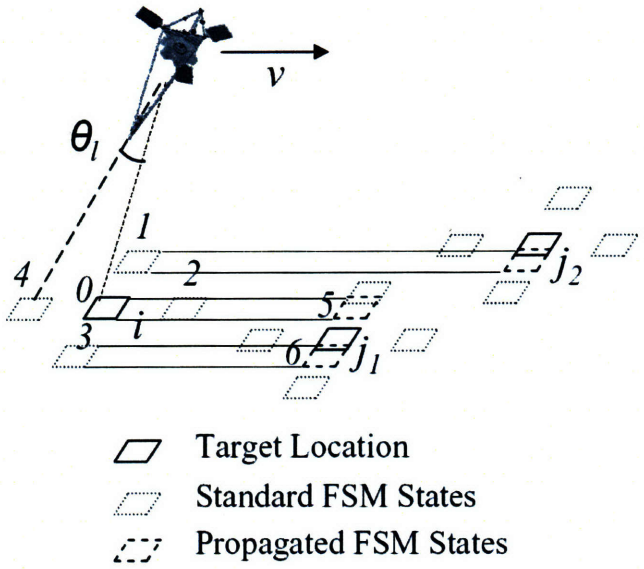


Figure 4-13: FSM Zero-Slew Propagated States

still valid, may not be required. The associated states may be left out for the sake of computational efficiency. If the range of the FSM is large, then many states may need to be propagated, significantly hampering the execution time of the algorithm.

4.5.2 Mixed-Integer Linear Programming

A Mixed-Integer Linear Programming (MILP) problem [9] is developed in order to try and alleviate the burden of discretizing a large continuous space, as in dynamic programming. This allows for a continuous range of FSM angles and imaging times. The overarching non-linear problem is linearized around specific nominal points to accomplish this task.

The basic framework of the problem is the same as in Section 4.4.2. The notions of a path through an array of targets with start and end constraints are still applicable. Where this approach diverges is in the handling of the dynamic constraints. Previously, two targets could only be imaged sequentially if there was sufficient time on orbit. This time limit was computed up front, based only on the location of the targets. In the formulation being pursued, the feasibility of imaging two consecutive targets is a function of several decision variables surrounding the process. This enables relaxation of the roll constraint and addition of the FSM capability.

Consider the problem geometry presented in Figure 4-14. Let p_i and q_i be the angular rotation of the FSM away from the telescope bore sight direction in the satellite pitch and

roll directions, respectively, while imaging target i. These equivalently represent the angular displacement of the satellite, in the pitch and roll directions, relative to the straight line of sight between the satellite and target. The imaging windows for each target are displayed on a time line below the figure. Note that t'_i represents the time when the satellite passes target i in the along-track direction (x), while t_i represents the time the image is taken.

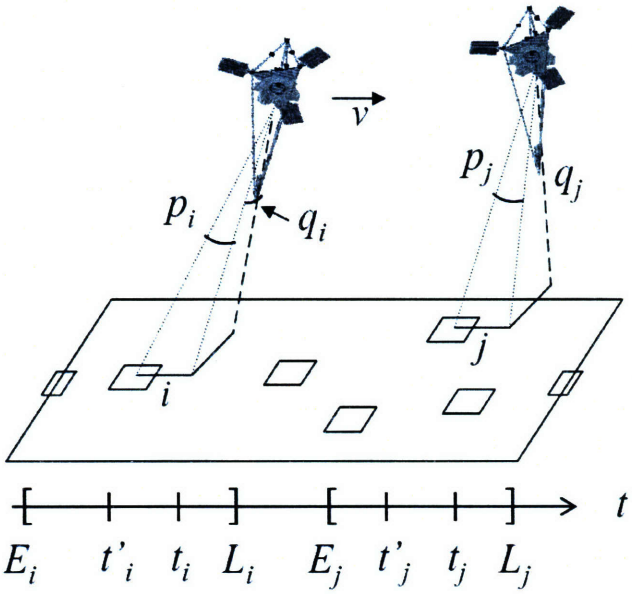


Figure 4-14: MILP Problem Geometry

In order to account for the more complicated imaging geometry, several variables are introduced. Let s_{ij} represent the nominal transition time, calculated as in Equation (4.1), and let $\bar{s}_{ij}$ be the modified transition time. Note that the true transition time s_{ij} is a strong non-linear function of t_i and t_j. This relationship for $\bar{s}_{ij}$ is linearized around the nominal s_{ij}. The variable z_{tij} is used to represent the difference in relative timing of two consecutive images:

$$z_{tij} = \left|(t_i - t'_i) - (t_j - t'_j)\right| \tag{4.7}$$

If images of targets i and j are both taken at the beginning of their respective imaging windows, then the orientation of the satellite in the pitch direction does not significantly change between the two images. The motion of the satellite is predominantly a slew in the roll direction, which is accounted for by s_{ij}. Since the value $t_i - t'_i \approx t_j - t'_j$, the value $z_{tij} = 0$. The value of z_{tij} is used as a measure of the required slew in the pitch direction

between consecutive images.

The values p_i, the pitch motion of the FSM, are assumed to be zero in this formulation. The parameter γ is a discount factor used to modify the value of off-axis imagery. Equation (4.8) shows the MILP formulation. Equations (4.3) are still binding.

$$\max \sum_i \left[w_i x_{ij} - \frac{\gamma}{fsm_{lim}} (z_{qi}) \right] \tag{4.8a}$$

$$t_i + d_i + \bar{s}_{ij} \leq t_j + M(1 - x_{ij}) \tag{4.8b}$$

$$s_{ij} + \frac{\partial s_{ij}}{\partial \phi_{ij}} \left[\alpha b_{ij} (q_j - q_i) + \kappa \frac{\partial \phi_{ij}}{\partial z_{tij}} z_{tij} \right] = \bar{s}_{ij} \tag{4.8c}$$

$$\left| (t_j - t'_j) - (t_i - t'_i) \right| \leq z_{tij} \tag{4.8d}$$

$$E_i \leq t_i \leq L_i \tag{4.8e}$$

$$\bar{s}_{ij} \geq 0 \tag{4.8f}$$

$$|q_i| \leq z_{qi} \tag{4.8g}$$

$$z_{qi} \leq fsm_{lim} \tag{4.8h}$$

Constraint (4.8b) is the transition feasibility constraint. It states that, if x_{ij} is chosen as a path, then the time at which i is imaged, plus the image duration d_i and the transition time $\bar{s}_{ij}$, must be less than the time at which j is imaged. The parameter M is an arbitrarily large constant (e.g. 10^6), which serves the purpose of relaxing the constraint when x_{ij} is not a chosen path. In that case, it is not necessary to meet the specific timing constraint associated with targets i and j. Constraint (4.8c) is the major modification from the problem previously described in Section 4.4.2. It utilizes a first derivative approximation of the function $f(\phi_{ij}) = s(\phi_{ij})$. The angular step size multiplier term, in brackets, is a combination of two elements. The first term comes from the fact that the satellite's bore sight may point slightly off the target range vector, increasing the angular change of ϕ_{ij}. The variable b_{ij} is binary, and set to one if $y_j > y_i$. This is necessary in establishing which direction reduces ϕ_{ij}. The final term accounts for deviations as a result of changes in imaging timing between two targets, to be described momentarily. Constraint (4.8d) defines the variable z_{tij} as the maximum timing difference between the two images i and j. If $z_{tij} = 0$, then the satellite does not have to be reoriented in the pitch direction. Constraint (4.8e) enforces the imaging window constraint and (4.8f) ensures that the transition time

remains positive. The remaining constraints (4.8g) through (4.8h) establish the limits of the FSM.

The term $\frac{\partial \phi}{\partial z_{tij}}$ is calculated as a conservative estimate based on the worst case geometries of i and j. Many options exist for determining this parameter. The most simple (but most constraining) is to assume that $t_i = t_i'$ and $y_i = y_j = 0$.

$$\frac{\partial \phi}{\partial z_{tij}} \approx \frac{v}{h} \tag{4.9}$$

The estimate is equivalent to the rotation rate of the target directly under the satellite. More complex relationships may be derived using a full model of the transition time based on Equations (4.5). The parameters α and κ are weights for the contribution to the satellite transition time from different sources. They are nominally set to one. Since the MILP model is conservative under these nominal settings, it is possible to adjust α and κ to search for a better solution. When this is done, a constraint check must be performed on the resulting optimal solution to ensure that the path is indeed feasible.

Several limitations of this approach should be noted. Because constraint (4.8c) is the result of linearizing a more complicated multi-variable function, some information is lost. The alternative is to solve a non-linear optimization problem, which is much more computationally difficult. The estimate of transition time is, however, conservative, so a feasible path is produced as an output.

Finally, when consecutive images are very close in y ($|y_j - y_i|$ small), the algorithm may predict maximal FSM usage for both images i and j to drive $\bar{s}_{ij}$ to zero. In these cases, the FSM is used to reduce the angle that must be traversed after image i is taken. When $|y_j - y_i|$ is small, the initial angle ϕ_{ij} is small. While reductions are possible using the FSM, eventually the effective angle $\bar{\phi}_{ij}$ will become zero and then negative. The constraints as written in Equation (4.8) assume that further reductions in transition time can be made, when in fact the angle ϕ_{ij} has gone through zero and is again increasing. This problem can be alleviated by adding more complicated constraints, though depending on the target density it may rarely be encountered.

4.5.3 Preliminary Analysis of FSM Performance

The dynamic programming formulation is implemented to find an estimate for the percent utilization of a FSM. This is done without zero-motion state propagation, so the results do not reflect the ability to rapidly acquire targets in close proximity using only the FSM. The same parameters are used as in Section 5.2, with $f(\phi)$ given by the f-number 1.5 satellite design. N is reduced to 20 to allow for the FSM angle to be sufficiently discretized. The maximum FSM range, $\theta_{l,max}$, is parameterized between 0° and 10°, and the FSM space is discretized to create five feasible angles including the nominal. The off axis penalty, γ, is set at 5%. An image of i taken using the FSM is therefore worth $0.95w_i$. Figure 4-15 shows the average percentage improvement averaged over 400 trial target sets, as a function of the FSM range.

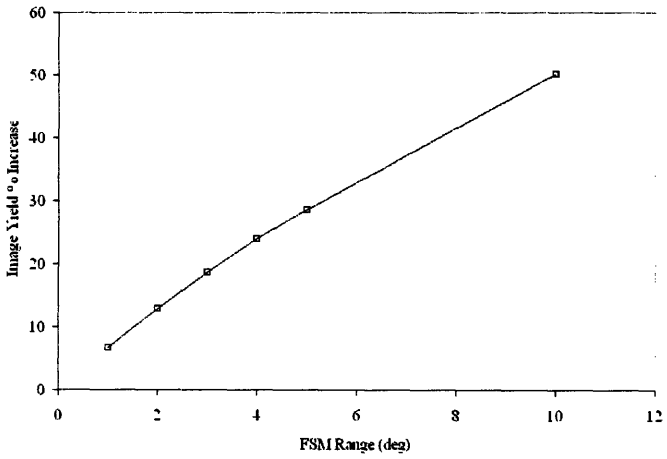


Figure 4-15: Added Value of Steering Mirror as a Function of Imaging Range

At a 500 km altitude orbit, a 3° range of the FSM yields a 19% improvement in the imaging productivity. A 10° range corresponds to approximately 176 km on the ground - approximately 10% of the 1732 km swath of the field of regard. The operational flexibility granted by the large range of motion enables a nearly 50% increase in imaging productivity. As discussed in Appendix C, a typical range for a FSM located behind the optics is on the order of 3°. Larger angular capabilities would likely have to be provided by large steering mirrors located in front of the optics.

4.6 Summary

Operational performance of satellite systems is a critical metric to be considered in evaluating different designs. For Earth-observing systems, a reasonable performance metric is the efficiency with which images are collected. For a given design, this efficiency can be estimated using a variety of optimization tools, including graph search, linear programming, and dynamic programming. Several optimization approaches were presented. Dynamic programming was shown to be most efficient by allowing a more general version of the problem to be solved in comparable time to more restrictive methods. Two methods were additionally presented to enable analysis of satellite systems with steering mirrors. Both algorithms allow the system designer to gain insight on the benefits of adding steering mirror technology.

Chapter 5 uses the optimization problem formulations discussed in this chapter to perform analysis on the imaging productivity of different satellite designs.

Chapter 5

Integration of Imaging Performance Metrics into MOST

Chapter 4 discussed several optimization methods that can be used to solve the Image Collection Optimization problem. This chapter discusses an approach for analyzing the imaging productivity of different satellite designs, based on their slew and settle properties. Methods for integration of this process into the MOST model are discussed.

5.1 Introduction

The objective of developing the tools presented in Section 4.4 is to integrate them into the MOST model being developed at the MIT Space Systems Laboratory. Previous work has been done to optimize the control architecture and mirror design [15]. The focus has primarily been on astronomical observations, but the capability exists to model Earth-observing systems. An important figure of merit for these systems is their ability to rapidly collect imagery. This ability is a strong function of the agility of the satellite, including the settle time associated with flexible dynamics. In order to evaluate an imaging productivity figure of merit for different systems, first the agility is characterized in a meaningful sense that can be generated by the MOST model. Second, an appropriate target density is chosen to approximate an operational scenario. Finally, each system is analyzed through this framework on a series of target sets.

5.1.1 Characterizing Satellite Agility in MOST

The agility of the satellite is formulated as a function $f(\phi)$. The argument ϕ is the angle that the satellite must traverse, and the function evaluates the total time required. This total includes the time needed to perform the slew and let vibrations settle. For a rigid body satellite, in which there is zero settle time, the function is shown in Equation (5.1) for a minimum time bang-bang maneuver:

$$f_{rigid} = 2\sqrt{\frac{2\phi I}{T}} \tag{5.1}$$

where ϕ is the angle to traverse, I is the moment of inertia of the satellite about the axis of rotation, and T is the maximum torque capability of the satellite attitude control system. For lightweight satellites with low frequency structural dynamics, the analysis is more complicated. In actuating the satellite via reaction wheels, there are two major sources of residual vibration. The first source is due to imbalances in the reaction wheels. As they are accelerated to provide torque, certain structural modes of the satellite may be excited. The second source is that the profile of torque may have frequency components that match vibration modes of the satellite. The excitation of low frequency modes, such as those resulting from the solar panels, tends to dominate as they require more time to damp out. Some analysis has been done to evaluate the MOST model slew and settle properties [14]. This, however, has been limited to considering one slew angle ϕ.

In order to characterize the performance of a design, we seek to find a function $f(\phi)$ as described above. This is done by simulating the slew of a satellite through a desired angle. Many different torque profiles may be used to accomplish this. One profile that helps to minimize jerk is a zero-slope initial and final condition slew with sinusoidal behavior. This approach is used here. Figure 5-1 shows the shape of such a maneuver for a 40 second slew. The zero-slope profile is not time-optimal for a rigid body satellite, but for a flexible system it provides sufficient performance by limiting excitation of the satellite's normal modes.

Reference [14] contains a full discussion of the models implemented to perform this analysis. The process is to construct a Finite Element Model (FEM) of a telescope based on design parameters such as f-number, mirror areal density, and telescope type (monolithic versus segmented). An example FEM for a segmented mirror system is shown in Figure 5-2. Analysis is run using the DOCS toolbox [10] and NASTRAN to create a state-space

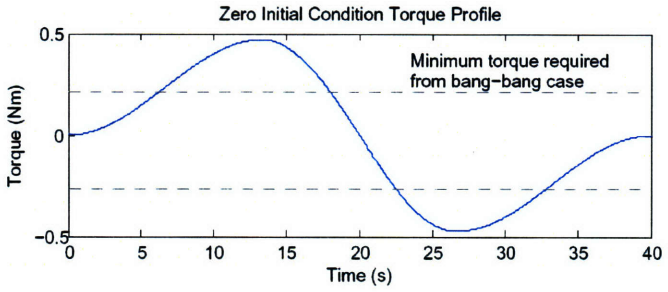


Figure 5-1: Zero Slope Initial and Final Condition Torque Profile [14]

model of the satellite dynamics. Attitude Control Systems (ACS) are added to the satellite to improve the dynamic properties. This integrated model is then used to analyze the performance of a slewing maneuver.

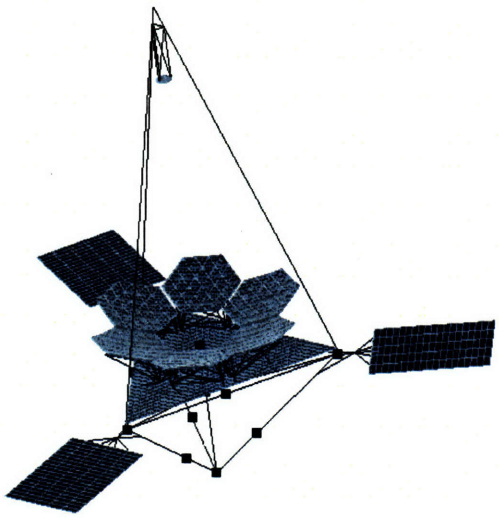
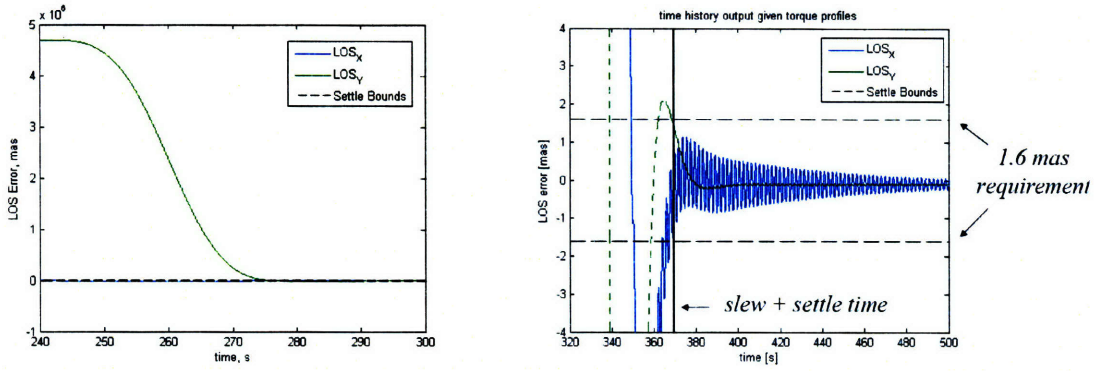


Figure 5-2: Sample FEM for a MOST Segmented Aperture System

A time simulation is performed, using this torque profile and the satellite model, to determine the time when the vibrations have settled to within the allowable Line of Sight (LOS) error. The total slew and settle time for this torque profile is recorded. The LOS error requirement used in this thesis is 1.6 mas (milli-arc seconds). Figure 5-3 illustrates the profile of a 5° angular slew with the LOS error settling to within requirements. Notice the different frequency content in the LOS_x and LOS_y directions. Higher frequency structural modes are excited in the LOS_y direction. The system settles to within requirements at approximately $t = 370$ seconds.



(a) Large Scale Slew Profile

(b) Close-Up of Settle Behavior

Figure 5-3: Calculation of Settle Time After a 5° Slew

An optimal control approach could also be used, but severely limits the automation of the trade analysis process without providing a huge benefit in the resulting settle time.

For each setting of angular change ϕ, the maneuver was simulated using several different slew times. Short slew times require large torque and induce large vibrations that must settle out. Long slew times induce smaller vibrations. A trade off must be made between the two components to find the optimal slew and settle time. Figure 5-4 illustrates this for a single satellite. A range of slew angles from two to thirty degrees is analyzed. For each slew angle, a torque profile is generated corresponding to a specific slew time, and the slew + settle time is determined. The minimum of each curve is a point on the function $f(\phi)$.

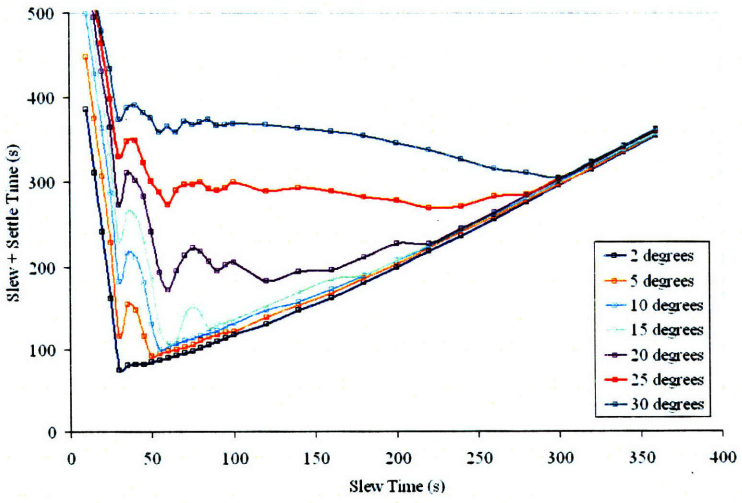


Figure 5-4: Performance of a Single Telescope Across Several Slew Angles and Slew Times

5.1.2 Comparison of Satellite Design Agility

Three designs are described here to illustrate the process. The first and second designs are segmented aperture systems with 10 kg/m^2 areal density mirrors. The differentiating factor between the second two designs is f-number. The third design is a rigid body satellite with a moment of inertia I and maximum torque capability T comparable to the flexible segmented aperture systems. The agility characterizations $f(\phi)$ for each is shown in Figure 5-5.

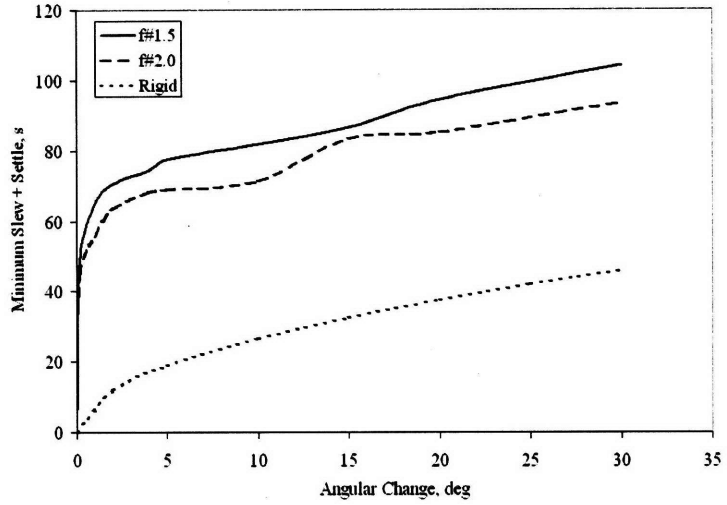


Figure 5-5: Function $f(\phi)$ Calculated for Three Designs: Two Flexible Systems and One Rigid Body System

Two observations can be made from this figure. First, the f-number 2.0 system provides shorter slew and settle times consistently across slew angles. Second, even very small attitude reorientation maneuvers require a significant settle time. This is very relevant when considering the use of a steering mirror to increase the field of regard, as is discussed in Section 4.5.

5.2 Sample Analysis

To analyze the imaging productivity of these designs, the dynamic programming formulation introduced in Section 4.4.3 is used. The simulation-based estimates for slew and settle time, with the dynamic programming problem formulation, allow for a fairly accurate picture of the capability of the satellites while remaining computationally tractable. Before running the analysis, several parameters of the simulation, including a, b, h, and η, are selected. The

values a = 10,000 km and b = 1,500 km are used. The minimum satellite elevation angle, η, is set at 30°. The altitude h is set as 500 km to represent a typical low-Earth orbit. The reward for capturing an image is a random integer between $w_i = 1$ and $w_i = 10$. Though this reward can be made a function of the imaging geometry, as discussed in Section 4.4.3, here it will remain independent of geometry.

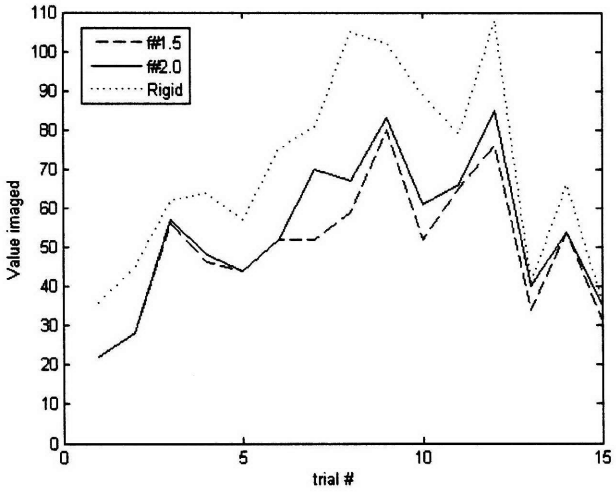


Figure 5-6: Small Set of Sample Trials for the Three Designs

Finally, a value for N is selected to represent an average scenario. The target density is set as approximately one target per $7*10^5$ km^2. Realistically it may be as high as one target per $7*10^3$ km^2. Nevertheless, the analysis highlights the effects of the different designs. Figure 5-6 shows the results for 15 trial target sets, comparing the two flexible satellites as well as an ideal rigid body satellite. Note the variability in optimal path value due to randomized target locations and priorities. Figure 5-7 shows the results for three satellite designs and three target density settings, where w_i is restricted to a value of one (all targets have the same priority). For each value of the number of targets N, 500 trials were run. The optimal path value was found using each of the three satellite agility functions $f(\phi)$. The optimal path values were then averaged for each satellite design.

As expected, the satellites with the shortest slew and settle time ($f(\phi)$) perform best. Note that increased target density produces larger comparative performance differences. This is to be expected. In a very target dense environment, small improvements in agility yield great performance rewards. The reduction in slew and settle time from the f-number 1.5 to the f-number 2.0 designs is approximately 11%, averaged over the range of slew angles.

This slew and settle time decrease results in a noticable increase in imaging productivity. The rigid satellite, with zero settle time, outperforms both flexible designs by more than 50% at high target densities.

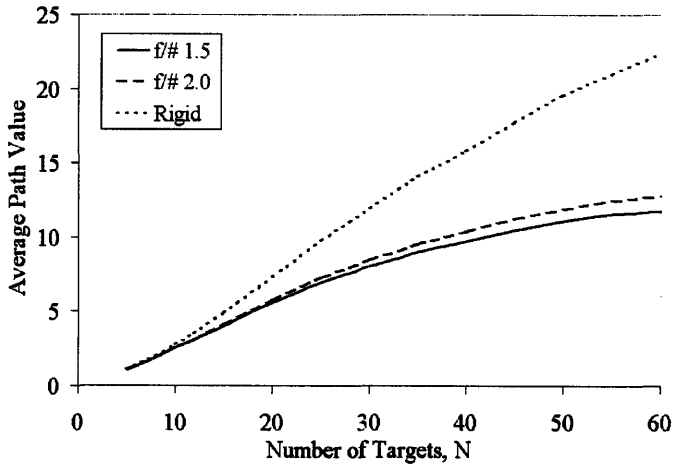


Figure 5-7: Average Optimal Path Value for Different Satellite Designs

Figure 5-8 shows the data in Figure 5-7 as a percentage of the total value of all targets in the scenario, which is $N-2$. It is clear that there is a maximum for each satellite as a function of the target density, which is proportional to N.

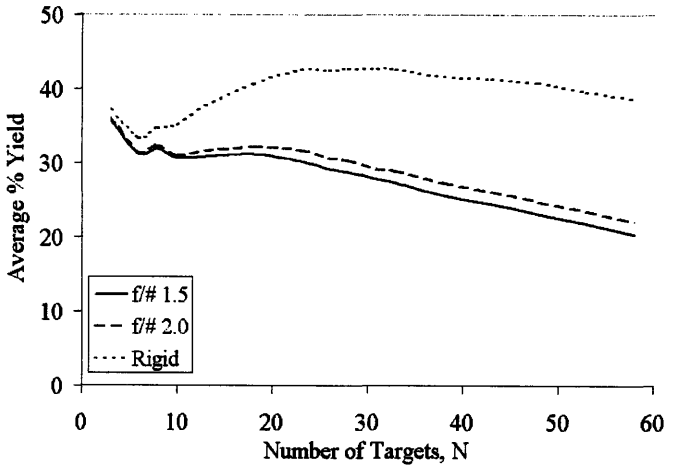


Figure 5-8: Percentage Yield of Optimal Paths for Different Satellite Designs

Variation of the altitude h and minimum target elevation angle η were not considered in this analysis. Adjustments in the orbital altitude affect orbital velocity, $v = \frac{\mu}{a}$. Over the range of LEO altitudes, however, v only changes by approximately 10%. The choice of η is largely be guided by the resolution requirements for images taken. If the satellite

system is required to maintain high resolution at a very low angle η, then the optics must be sized accordingly. At a constant target density, decreasing η increases the amount of targets available to be imaged, potentially enabling higher value imaging sequences. This is a second order effect, behind the fundamental agility of the satellite.

5.3 Parameterization of $f(\phi)$

While the optimization formulations utilized in Chapter 4 are sufficient for analyzing single problem instances, difficulty is encountered when trying to determine system performance over many random problems. The MOST model can analyze in MATLAB® a single telescope instantiation in approximately five minutes on a Pentium 4 3.0 GHz computer with 1.0 MB of RAM. Using a reduced dynamics model, a slew and settle time simulation takes approximately 30 seconds. Generating the function $f(\phi)$ experimentally for a single telescope requires finding the minimum slew and settle time, as a function of slew time, for each slew angle. An accurate determination of $f(\phi)$ may require 50-100 time simulations. After $f(\phi)$ has been generated, the imaging performance is estimated based on a series of simulations over different target sets. Very high target densities create large optimization problems. While the algorithms described can solve single instances of large problems, determining average performance over many data target sets is very resource intensive. This is very limiting in terms of the overall model efficiency.

An alternative approach is to parameterize the function $f(\phi)$ using a basis function $f_0(\phi)$ and scaling factor k such that, for any satellite design, $f(\phi) \approx k * f_0(\phi)$. There has been some indication that the shape of $f(\phi)$ is consistent across different satellite designs, and there is potential to exploit this to interpolate between different designs. The MOST model performs a small set of slewing simulations to approximate the performance of the given design for a single slew angle. This data allows approximation of the value k, and extrapolation across all slew angles by assuming the form $f(\phi) = k * f_0(\phi)$. The performance results are looked up in a pre-computed database based on the value of k. The agility of a satellite is a very clean interface between the complicated nature of the satellite design and the performance in operation. The parameterization of $f(\phi)$ can utilize this notion to eliminate the need to recompute imaging productivity metrics for each satellite design.

5.4 Summary

The MOST model, developed at the MIT Space Systems Laboratory, enables rapid trade-space exploration for lightweight flexible space telescopes. Previously, the model utilized slew and settle time as an operational performance metric. An approach was presented to find more accurate performance metrics by using appropriate optimization problem formulations, developed in Chapter 4, alongside a characterization of the satellite agility. A comparison between three systems demonstrated how changes in structural flexibility, and thus agility, affect the overall imaging performance.

Computational difficulties are seen in numerically determining the satellite agility function $f(\phi)$ and in calculating imaging performance metrics. Parameterization of the agility function $f(\phi)$ may alleviate some of the computational difficulties and allow practical integration into the MOST model. With this improvement, more accurate operational performance metrics can be included for consideration in the model trade space, increasing the utility of the MOST software for analyzing future lightweight space telescopes.

5.5 Future Work

There are several areas in which this research can be expanded.

- Improved Algorithms for Image Collection Planning: Optimization is a very expansive field, and there are many possibilities for pursuing more advanced algorithms to estimate imaging productivity metrics.

- Imaging Scenarios: The analysis presented assumed a series of point targets with constant priority and random location. There is potential to investigate the effects of more complex scenarios with, for example, clustered target locations or large area targets. Additionally, assigning an image priority based on both the target priority and the image resolution (from geometry) may affect the comparative imaging productivity of different satellite designs.

- Heuristics: The analysis presented in this chapter may hold potential for developing analytical relationships between satellite agility and performance which can be used in MOST.

- Direct Methods: Fields such as stochastic programming are able to analyze optimization problems with random variables. Application of these methods to randomized target locations may allow for direct computation of performance metrics instead of through simulation.

- Comprehensive Satellite Scheduling Formulations: This analysis only looked at satellite agility constraints. There are many other constraints, including power, thermal, and data storage, which may constrain satellite operations. A more complete satellite model could be run through a scenario analysis to find the most violated constraints. This information could be used to adjust the satellite design accordingly.

Chapter 6

Conclusions

This chapter contains a summary of the work presented in this thesis, final conclusions, and contributions.

6.1 Thesis Summary

Reliance on space resources has increased significantly over the past several decades. Demand for Earth imagery from the scientific and military communities has steadily grown. Commercial parties have also become very interested imagery for a variety of purposes, from urban planning to interactive map making. Satellites that are built to provide Earth imagery for these organizations tend to have the objectives of 1) efficient coverage of a changing global database of targets and 2) responsive coverage of regional real-time events. The importance of each objective depends on the specific application.

This thesis investigated two aspects of Earth-observing satellite systems. Chapter 2 introduced a satellite constellation architecture designed to support both coverage and responsiveness objectives. The RECON architecture features a nominal global observing mode constellation to support long term imaging objectives, along with the capability to reconfigure into a more focused regional observing mode. Several relevant performance parameters were defined, including regional observing mode access time and constellation responsiveness. A cost model was developed to support analysis of many different constellations within the RECON architecture.

Analysis of a set of equal-cost satellite constellations suggests that the lowest altitudes are optimal. Optical requirements strongly influence the cost of each vehicle, and reducing

altitude releases money from the program budget to be spent on building additional satellites or increasing the field of regard.

In the example mission described, the satellites are designed for a ten year mission with two reconfiguration maneuvers per year. To support the reconfiguration objective for a system at 554 km altitude, an additional 200% ΔV is needed over the stationkeeping and end-of-life disposal ΔV requirements. While this is significant, the propellant mass typically remains less than 30% of the total wet mass. Stationkeeping requirements at very low altitudes are excessive. Architectures presented at the low 262 km altitude push the boundaries of the model due to the enormous propellant mass needed.

The coverage of a specific target of interest improves two to three times when reconfiguring the satellite constellation from the global to the regional observing mode. For a moderately sized constellation in global observing mode (15 - 20 vehicles), upon notification of a target of interest the first image can generally be taken within several hours. The first vehicle can be placed on the appropriate repeating ground track within one day. The response time of the constellation, which is defined as the time until the first regional observing mode image is taken, is on the order of one day.

Chapter 3 introduced performance metrics for the global observing mode constellation to increase the computational efficiency of the problem. After developing an appropriate objective function, the optimal constellation results obtained are similar to those in Chapter 2. Comparisons are made to a fleet of UAVs designed to fulfill the objective of responsiveness. Based on several example UAV architectures, the cost and performance are comparable to a satellite constellation, though different benefits are offered. The major advantage of UAVs is the persistence that can be achieved once the target is reached. For the purposes of surveying a natural disaster, UAVs are very well suited. For intelligence purposes on the other hand, space systems offer a much greater reliability at lower risk.

Chapter 4 introduced the Image Collection Optimization problem and discussed three approaches for solving it. A graph search method is easily implemented, and enhancements can be made to improve the search efficiency by pruning inferior paths during the search process. Linear programming is a second method to find optimal paths based on the Traveling Salesman Problem. A dynamic programming algorithm is also applied. Numerical experiments indicate that dynamic programming is most efficient. Two approaches were discussed for adding a steering mirror capability to the optimization process.

Chapter 5 used these optimization formulations to analyze the performance of several different satellite designs. Each satellite configuration has different dynamic properties, resulting in different slewing agility properties. The agility properties can be easily captured by a function $f(\phi)$ that describes the slew and settle time for different slew angles. Each satellite is characterized in terms of an agility function, with structural flexibility requiring substantial settling time. This information is used to evaluate the imaging productivity of the different satellite designs over a set of randomly located targets. Calculation of $f(\phi)$ and image collection optimization algorithms, however, present computational difficulties. Recommendations are made for parameterizing the agility function f to improve computation time, and pursuing application of stochastic programming as a more direct method.

6.2 Contributions

The contributions of this thesis are summarized below.

- Developed a Responsive Space architecture to support both long term imaging and responsiveness objectives.
- Analyzed a representative set of RECON satellite constellations to find optimal configurations.
- Compared space-based and air-based responsive architecture alternatives.
- Implemented three satellite image collection optimization algorithms to enable comparative analysis of different satellite designs.
- Examined the effects of satellite agility, structural flexibility, target density, and steering mirror capability on imaging productivity.

6.3 Future Work

Please reference Sections 2.9 and 5.5.

Appendix A

Walker Constellations

A.1 Coverage Distribution

The nature of Earth observing satellites is that coverage is maximum near the inclination of the orbit. To evaluate the disparity between coverage at the equator, and coverage near the inclination, a series of constellations are evaluated. Figure A-1 shows the percent coverage per unit time of a target at varying latitudes.

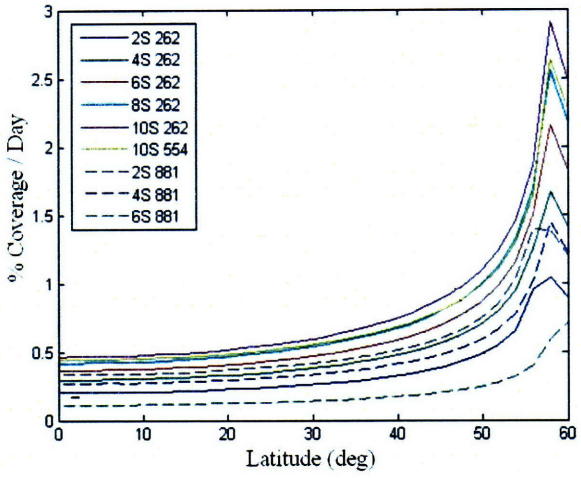


Figure A-1: Coverage of Different $i = 60°$ Walker Constellations Across Latitude

The coverage of a given target is a strong function of the target latitude. Section 3.1.2 developed a metric for approximating the coverage of a Walker constellation. To calibrate this metric to provide performance at the Equator, we must consider how the coverage is distributed across latitude.

The average value of coverage is first determined by integrating each of the curves in Figure A-1 and dividing by the latitude range (60°). The ratio of performance at 0° latitude to the average is defined as ξ. Based on the information contained in Figure A-1, ξ is approximated as 0.55. That is, for a typical constellation, the coverage at the equator is 55% of the average value.

A.2 Metric Validation Against STK Simulation

The metric M_2 for global coverage, in Equation (3.5), is validated against a set of simulations in STK^{TM}. The set used is the same as shown in Table 2.5. Figure A-2 shows the comparison between the M_2 approximation and the STK^{TM} simulation results for 16 different constellations. The average error is approximately 3%, while the standard deviation in the error is 16.5%.

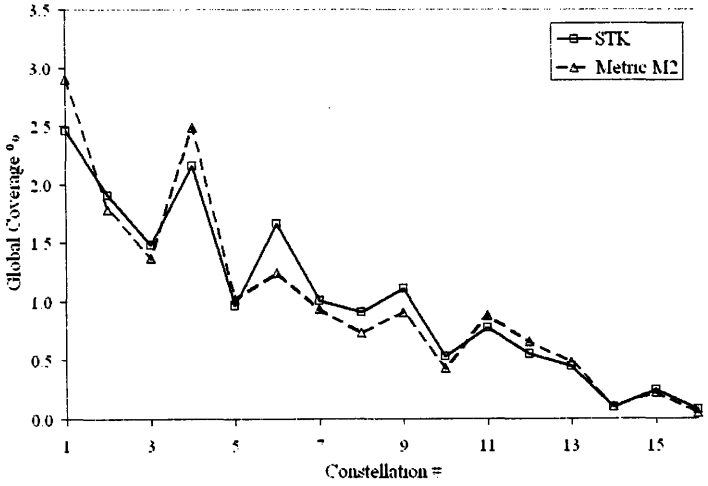


Figure A-2: Validation of Metric M_2

Appendix B

Graph Search Methods

To illustrate the differences between search methods, consider the example graph in Figure B-1.

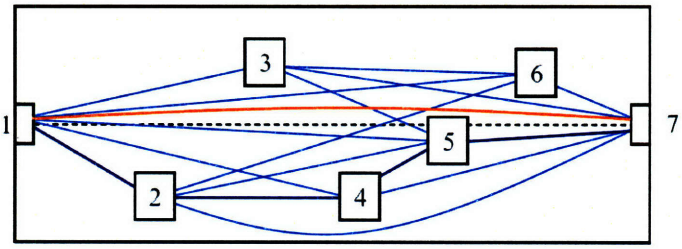


Figure B-1: Example Graph

This graph has $N = 7$ nodes, which comprise the set of vertices V. The arcs from the set E are drawn between nodes (i, j) where the transition is feasible. The search tree associated with this graph is shown in Figure B-2. This tree represents all feasible paths from node one to node N. There many different ways to perform a search of this graph for the best solution, the most basic of which are Depth-First and Breadth-First.

Depth-First explores the tree by adding newly expanded paths to the top of the search queue, while Breadth-First adds newly expanded paths to the bottom of the search queue. Given a selection of paths to expand, all at the same depth in the search tree, both methods explore the nodes in ascending order. Breadth-First places search priority on paths that, in the search tree, are close to the start node (at a low depth). Depth-First prioritizes searching for the goal node deep in the search tree. Each method has advantages based on the specific application. In general, Depth-First search is more computationally intensive, while Breadth-First is more memory intensive. Reference [54] contains a comprehensive

discussion of graph search algorithms.

In this application, the ordering of the nodes from left to right means that Depth-First search will find solutions that visit many nodes before reaching the goal node. Although the priority, w_i, of these nodes may be low, it is generally the case that a path with many nodes has a higher value. In this sense, Depth-First is advantageous. The first discovered path has potential to be a very good solution, and enables efficient pruning as the search continues.

Breadth-First search, for this application, first finds paths to N that visit the fewest number of nodes. As a result, future solutions tend to be incrementally better as more and more nodes are visited. This aspect reduces the efficiency with which poor solutions are pruned from the search.

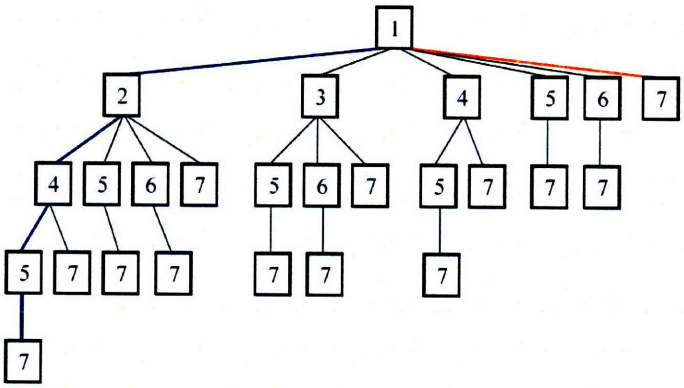


Figure B-2: Example Search Tree

Appendix C

Cassegrain Mirror Geometry

For a diffraction limited Cassegrain mirror design with a parabolic primary mirror, the mirror shape is defined by Equation (C.1):

$$y = \frac{z^2}{4f} \tag{C.1}$$

$$\theta_m = arctan(\frac{z}{2f}) \approx \frac{z}{2f} \tag{C.2}$$

$$d = \frac{f}{2} \tag{C.3}$$

where y is the radial direction, z is the out-of-plane direction, f is the focal length, and θ_m is the slope of the mirror. Figure C-1 shows an outline of the geometry of off-axis light entering the telescope. After being reflected by the primary and secondary mirrors, the light arrives at a distance Δr from the nominal central location. Geometry indicates that this Δr is related to the off-axis angle θ_l by Equation (C.4).

$$\Delta r = d\,(tan(\theta_m + \theta_l) - tan(\theta_m) + tan(\theta_l)) \approx 2d\theta_l \tag{C.4}$$

The physical limit imposed here is that Δr must be sufficiently small such that the light still reaches the focal plane to be captured by the CCD. Assume that Δr_{max} is given as a fraction of primary mirror diameter D. The maximum off axis angle is given by Equation (C.5):

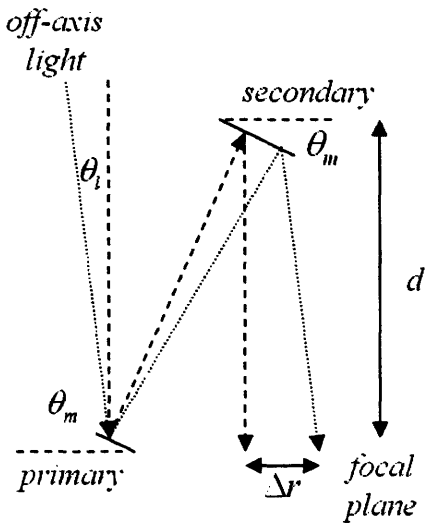


Figure C-1: Geometry of a Two Mirror Telescope Configuration

$$\theta_{l,max} = \frac{\Delta r_{max}}{f} = \frac{kD}{f} = \frac{k}{f/\#} \tag{C.5}$$

Here, $f/\#$ is the f-number of the mirror, the ratio of the focal length to diameter, f/D. For an $f/\#$ 2.0 system with $k = 0.1$, the mirror can focus light up to $\theta_{l,max} = 2.9°$.

Appendix D

ReCon Source Code

This appendix contains the source code used in Chapter 3.

```
1  function xopt = reconOptimization(BUDGET,orbit,maxDelta_a,res,alpha,beta)
2
3  %% %%%%%%%%%%%%%%%%%%%%%%%%
4  % reconOptimization.m
5  % %%%%%%%%%%%%%%%%%%%%%%%%
6  %
7  % This function explores the ReCon tradespace to find optimal designs.
8  %
9  %   INPUTS   BUDGET          Program budget. [dollars]
10 %            orbit           Orbit altitude selection. [1-3]
11 %            maxDelta_a      Maximum altitude offset. [km]
12 %            res             Resolution of trade-space. [ND]
13 %            alpha           Objective function weights. [ND]
14 %            beta            Objective function balancing parameter. [ND]
15 %
16 %
17 %   OUTPUTS  xopt       Design Vector for optimal design.
18 %                             x(1): N
19 %                             x(2): theta, rad
20 %                             x(3): D, m
21 %                             x(4): delta-a, km
22 %                             x(5): m_dry, kg
23 %                             x(6): m_p, kg
24 %                             x(7): delta-v-recon, m/s
```

```
25 %                              x(3): Δ-v-total, m/s
26 %%
27
28 global plotFlag
29 global dispFlag
30 rtd = 180/pi;
31
32 %-------------------------------
33 % Orbit Parameters Look Up
34 %-------------------------------
35
36 % Table of orbit parameters
37 % k | h | rho
38 design_table = [16, 262, 6.24e-11;
39                 15, 554, 2.21e-13;
40                 14, 881, 4.66e-15];
41 % k    %ND          Number of revolutions/repeat
42 % h    %km          Orbital Altitude
43 % r_a  %km          Orbital Radius
44
45 % Retrieve parameters based on 'orbit' input variable
46 k   =   design_table(orbit,1);
47 h   =   design_table(orbit,2);
48 rho =   design_table(orbit,3);
49
50 %-------------
51 % Constants
52 %-------------
53 R_e = 6378;          %km          Earth Radius
54 mu = 3.98*10^5;      %km^3/s^2    Gravitational Parameter
55 v_0 = 300*9.81;      %I_sp*g      Exhaust Velocity
56 lam = 750e-9;        %m           Observing Wavelength
57 d_x = 1;            %m           Ground Resolution Requirements
58 ball = 200;         %kg/m^2      Ballistic Coefficient
59 L_m = 10;           %years       Mission Lifetime
60 N_r = 2;            %            Number of Reconfigurations
61 r_inj = R_e + 185;  %km          Injection Altitude
62 r_disp= R_e + 50;   %km          Disposal Altitude
63
```

```
64 %-------------------------
65 % Cost Model Parameters
66 %-------------------------
67 c_optics = 15.02*10^6;          %dollars/m^2.6    Optics Costs
68 c_dry = 0.113*10^6;             %dollars/kg       Spacecraft Bus Costs
69 c_launch = 15280;               %dollars/kg       Launch Costs
70 c_ops = 0.03*L_m;               %dollars/SC       Operations Cost/SC
71 c_m = 653.6/(0.7)^3;            %kg/m^3      Dry Mass Scaling
72 c_p = 1.15;                     %ND          Propellant Infrastructure Mass
73
74 %-------------------------
75 % Derived Parameters
76 %-------------------------
77
78 % Orbital radius [km]
79 r_a = R_e + h;
80 % Orbital period [years]
81 T = 2*pi*sqrt(r_a^3/mu)/(365*24*60*60);
82 % Stationkeeping Δ-v requirements [m/s/year]
83 dV_s = 10^6*pi*1/ball*rho*r_a*sqrt(mu/r_a)/T;
84 % Injection Δ-v requirements [m/s]
85 dV_inj = 10^3*sqrt(mu)*(abs(sqrt(2/r_a-2/(r_a+r_inj)) ...
86     - sqrt(1/r_a)) + abs(sqrt(2/r_inj-2/(r_a+r_inj)) ...
87     - sqrt(1/r_inj)) );
88 %   Disposal Δ-v requirements [m/s]
89 dV_d = sqrt(mu)*abs(sqrt(2/r_a-2/(r_a+r_disp)) - sqrt(1/r_a))*10^3;
90 D_min = 1.22*lam*h*10^3/d_x;     %m       Mirror diameter
91 m_min = c_m*D_min^3;             %kg      Minimum bus mass
92 th_min = 0;                      %rad     Min off-nadir angle
93 th_max = asin(R_e/(R_e+h));      %rad     Max off-nadir angle (horizon)
94
95 %-------------------------
96 % Define Trade Space
97 %-------------------------
98
99 x2 = linspace(th_min,th_max,res);     % Range of off-nadir angle theta
100 x4 = linspace(0,maxDelta_a,res);      % Range of altitude offset Δ-a
101 f = zeros(length(x2),length(x4)); opt = 0;
102
```

```
% Loop over off-nadir angle
for i2 = 1:length(x2)
    % Loop over altitude offset
    for i4 = 1:length(x4)
        theta = x2(i2); d_a = x4(i4);
        % Calculate coverage parameters
        [K,arc,range] = footprint(theta,h);

        % Create Design
        D = 1.22*lam*range/d_x*10^3;    % Calculate diameter
        m_dry = c_m*D^3;                % Calculate dry mass
        if m_dry < 200
            m_dry = 200;
        end
        dV_r = 0.5/r_a*sqrt(mu/r_a)*10^3*d_a;    % Recon Δ-V
        dV_m = 2*N_r*L_m*dV_r + dV_s*L_m;        % Mission Δ-V
        m_disp = m_dry*(exp(dV_d/v_0)-1);        % Disposal propellant
        m_p = (m_dry+m_disp)*(exp(dV_m/v_0)-1); % Mission propellant
        m_p = m_disp + m_p;                      % Updated propellant
        dV_T = dV_m + dV_d;                      % Total Δ-V
        m_inj = (m_dry+m_p)*(exp(dV_inj/v_0)-1);% Injection propellant

        % Calculate Cost
        J = [c_optics*D^2.6;                              % Optics
             c_dry*m_dry;                                 % Bus
             c_launch*(m_dry + m_p + c_p*m_inj)];         % Launch
        J(4) = c_ops*sum(J(1:2));                         % Operations

        % Select number of satellites
        cost_1 = sum(J(1:2)); cost_2 = sum(J(3:4));
        N = floor(learningCurve(cost_1,cost_2,BUDGET));

        % Calculate Objective Function Value
        x = [N,theta,D,d_a,m_dry,m_p,dV_r,dV_T];
        temp = -objectiveFunction(x,0,orbit,alpha,beta);
        f(i2,i4) = temp;

        if temp > opt
            opt = temp;
```

```
142             xopt = x;
143         end
144
145     end
146 end
147
148 %-------------------------
149 % Display Optimal Design
150 %-------------------------
151 if dispFlag
152     disp(['N Satellites:    '    num2str(xopt(1))]);
153     disp(['Theta, deg:      '    num2str(xopt(2)*180/pi)]);
154     disp(['D, m:            '    num2str(xopt(3))]);
155     disp(['Δ-a, km:     '    num2str(xopt(4))]);
156     disp(['Dry Mass, kg:    '    num2str(xopt(5))]);
157     disp(['Prop Mass, kg:   '    num2str(xopt(6))]);
158     disp(['dV, Recon, m/s:  '    num2str(xopt(7))]);
159     disp(['dV, Total, m/s:  '    num2str(xopt(8))]);
160     objectiveFunction(xopt,1,orbit,alpha,beta);
161 end
162
163 if plotFlag
164     figure; surf(x4,x2*rtd,f);
165     xlabel('Delta-a, km')
166     ylabel('\theta, deg')
167     zlabel('Q')
168
169     figure; [C,h] = contour(x4,x2*rtd,f);
170     set(h,'ShowText','on','TextStep',get(h,'LevelStep')*2)
171     xlabel('Delta-a, km')
172     ylabel('theta, deg')
173     zlabel('Q')
174 end
```

```
function [f,M] = objectiveFunction(x,dispFlag,orbit,alpha,beta)

%% %%%%%%%%%%%%%%%%%%%%%%%%
% objectiveFunction.m
% %%%%%%%%%%%%%%%%%%%%%%%%
%
% This function returns the objective function value of a ReCon design.
%
%   INPUTS  x           Design Vector for optimal design.
%                           x(1): N
%                           x(2): theta, rad
%                           x(3): D, m
%                           x(4): Δ-a, km
%                           x(5): m_dry, kg
%                           x(6): m_p, kg
%                           x(7): Δ-v-recon, m/s
%                           x(8): Δ-v-total, m/s
%           dispFlag    Internal display flag.
%           orbit       Orbit selection, same as in reconOptimization().
%           alpha       Objective function weights.
%           beta        Objective function balancing parameter.
%
%   OUTPUTS f           Objective function value.
%           M           Vector of metric values.
%%

%-----------------------------
% Orbit Parameters Look Up
%-----------------------------

% Table of orbit parameters
% k | h | rho
design_table = [16, 262;
                15, 554;
                14, 881];

% Retrieve parameters based on 'orbit' input variable
k = design_table(orbit,1);
h = design_table(orbit,2);
```

```
40
41 %-----------
42 % Constants
43 %-----------
44 R_e = 6378;          %km          Earth Radius
45 mu = 4*10^5;         %km^3/s^2    Gravitational Parameter
46 Day_s = 86200;       %s           Length of Sidereal Day
47 C = 0.66;           %ND          Coefficient for M2 = Xi / sin(i)
48 r_a = 6378 + h;     %km          Orbital Radius
49
50 [K,arc] = footprint(x(2),h);
51
52 % Metric Value Calculation
53 M = [ 100*x(1)*arc/(k*pi*R_e);                    % Regional Coverage
54       100*x(1)*K/(4*pi*R_e^2)*C;                  % Global Coverage
55       x(1)*3*pi/Day_s*sqrt(r_a/mu)*k^2*x(4)];     % Response Time
56
57 % Construct Objective Function
58 q1 = -diag(alpha)*M;
59 Q = [q1; beta*abs(diff(q1))];
60 f = sum(Q);
61
62 if dispFlag
63     disp(['Objective Value: ' num2str(f)])
64     disp('Reg Coverage | Global Coverage | Response Time^-1 | Diff1 | Diff2');
65     disp([num2str(M(1)) '    ' num2str(M(2)) ...
66         '    ' num2str(1/M(3)) '    ' num2str(Q(4:5)')])
67 end
```

```
function N = learningCurve(cost_1,cost_2,P)

%% ************************
% learningCurve.m
% ************************
%
% This function calculates the production learning curve for satellites.
%
%   INPUTS  cost_1      Satellite costs. [dollars]
%           cost_2      Launch + operations costs. [dollars]
%           P           Program budget. [dollars]
%
%   OUTPUTS N           Number of satellites affordable.
%
%%

% Learning curve parameters
A = 1.5025; B = 0.6442;

% Initial guess
x0 = (P/(cost_1 + cost_2))^(1/B);
% Solve for the maximum number of satellites that can be purchased
N = fzero(@(x)A*x^B*cost_1 + x*cost_2 - P,x0);
```

```
function [K,arc,range] = footprint(theta,h)

%% %%%%%%%%%%%%%%%%%%%%%%%%
% footprint.m
% %%%%%%%%%%%%%%%%%%%%%%%%%
%
% This function calculates observation geometry for a satellite.
%
%   INPUTS  theta       Off-nadir angle. [deg]
%           h           Orbital altitude. [km]
%
%   OUTPUTS K           Footprint surface area. [km^2]
%           arc         Range along Earth surface, nadir to target. [km]
%           range       Range to target (h'). [km]
%%

R_e = 6378; %km     Earth Radius

% Calculate relevant geometric parameters
eps = acos(sin(theta)*(R_e/(R_e+h))^-1); %rad   Elevation angle
lam = pi/2 - theta - eps;                %rad   Earth central angle
arc = lam*R_e;                           %km    Range from nadir to target
b = R_e - R_e*cos(lam);                  %km    Parameter for area calc.
K = 2*pi*R_e*b;                          %km^2  Viewable Area

% Avoid singularity in calculation of range
if theta < 0.05
    range = h/cos(theta);
else
    range = R_e * sin(lam)/sin(theta);
end
```

Appendix E

Image Collection Optimization Source Code

This appendix contains a subset of the source code used in Chapters 4 and 5.

```
1  function [optPath,m_extendedPathsCount,runTime] ...
2              = depthFirstBranchBound(x,y,w,n,agilityTable,SATELLITE)
3
4  %% %%%%%%%%%%%%%%%%%%%%%%%%
5  % depthFirstBranchBound.m
6  % %%%%%%%%%%%%%%%%%%%%%%%%
7  %
8  % This function finds the optimal path for imaging a series of ground
9  % targets using a graph search algorithm.
10 %
11 %  INPUTS   x            Vector of target x-coordinates (length n). [km]
12 %           y            Vector of target y-coordinates (length n). [km]
13 %           r            Vector of target priority (length n). [ND]
14 %           n            Number of targets.
15 %           aT           Lookup table containing satellite agility
16 %                        information; each row is an [angle(deg), time(s)]
17 %                        data point, the time required for the
18 %                        satellite to slew the given angle.
19 %           SATELLITE    Structure containing satellite and imaging area
20 %                        specifications. See 'Satellite Parameters' below.
21 %
```

```
22 %  OUTPUTS  optPath     Structure containing optimal path and cost.
23 %                           .path   Vector of targets to visit
24 %                           .cost   Optimal path value
25 %           runTime     Run time of the function call. [s]
26 %           m_extendedPathsCount:
27 %                       Number of paths that have been extended
28 %                       using getExtendedPaths().
29 %%
30
31 tic
32
33 % Set global variables
34 global dispFlag
35 global plotFlag
36 global m_extendedPathsCount
37 global m_pathsPruned
38
39 % Initialize counters
40 m_extendedPathsCount = 0;
41 m_pathsPruned = 0;
42 m_maxQueueSize = 1;
43
44 %-------------------------------
45 % Satellite Parameters
46 %-------------------------------
47 v = SATELLITE.v;     % orbital velocity
48 h = SATELLITE.h;     % orbital altitude
49 rtd = 180/pi;        % conversion rad to deg
50
51 % Normalize x locations to begin at zero
52 x = x-x(1);
53
54 % Create matrix of valid edges based on transition feasibility
55 edge = zeros(n);
56 for i = 1:n
57     for j = i+1:n
58         dp  = abs(atan(y(i)/h) - atan(y(j)/h));
59         bit = agilityLookup(dp*rtd,agilityTable) < abs((x(j)-x(i))/v);
60         edge(i,j) = bit;
```

```
61      end
62  end
63
64  %-------------------------
65  % Perform Graph Search
66  %-------------------------
67
68  % Initialize search queue, optimal path, and pruning vector c
69  Q = struct('path',{1},'cost',0);
70  optPath = struct('path',{},'cost',-1);
71  c = zeros(1,n);
72  optCost = -1;
73
74  % While search queue is not empty
75  while (isempty(Q) ~= 1)
76      % If the goal node (N) is found
77      if (Q(1).path(1) == n)
78          % If the path is better, update optimal path
79          if Q(1).cost > optCost
80              optPath = Q(1);
81              optCost = optPath.cost;
82              if dispFlag > 1
83                  disp('optPath Updated:')
84                  disp(optPath.path);
85              end
86          % Else if the path is the same value, add it on
87          % to the list of optimal paths
88          elseif Q(1).cost == optCost
89              optPath = [Q(1) optPath];
90          end
91      % Else if the goal node is not found, keep exploring
92      else
93          % Find all possible paths to extend from the current path
94          [extendedPaths,c] = getExtendedPaths(Q(1),w,edge,c);
95          % If new paths are found, add them to the queue
96          if (isempty(extendedPaths) == 0)
97              % Breadth first search
98              % Q = [Q extendedPaths];
99              % Depth first search
```

```
100             Q = [Q(1) extendedPaths Q(2:end)];
101         end
102         % Update the maximum queue size
103         if length(Q) > m_maxQueueSize
104             m_maxQueueSize = length(Q);
105         end
106     end
107     % Clear the path that has just been expanded
108     Q(1) = [];
109 end
110
111 runTime = toc;
112 clear i j edge Q
113
114 if dispFlag > 0
115     disp(['Extended Paths: ' num2str(m_extendedPathsCount)])
116     disp(['Pruned Paths: ' num2str(m_pathsPruned)])
117     disp(['Max Queue Size: ' num2str(m_maxQueueSize)])
118     disp(['Runtime = ' num2str(runTime) ' seconds'])
119 end
120
121 if plotFlag
122     % Calculate corresponding image times
123     imgTime = x(optPath(1).path)/v;
124     plotPath(x,y,optPath(1).path,optPath(1).cost,imgTime,SATELLITE)
125 end
```

```
1  function [extendedPaths,c] = getExtendedPaths(pathStruct,w,edge,c)
2
3  %% ####################
4  % getExtendedPaths.m
5  % ####################
6  %
7  % This function calculates all the extended paths of the provided path.
8  %
9  % INPUTS:   pathStruct      A struct containing the path and associated cost.
10 %           w               Target set priority w (length N).
11 %           edge            Matrix containing feasible transitions.
```

```
12 %           c              Best discovered path vector.
13 %
14 % OUTPUTS:  extendedPaths  A struct containing all extended paths from
15 %                          path.
16 %           c              Updated best discovered path vector.
17 %%
18
19 global m_extendedPathsCount
20 global m_pathsPruned
21
22 pruning_flag = 1;
23 head = pathStruct.path(1);
24 len = length(edge);
25 extendedPaths = struct('path',{},'cost',{});
26
27 for i = head+1:len
28     if(edge(head,i) == 0)
29         continue;
30     end
31     if pruning_flag && (pathStruct.cost + w(i) < c(i))
32         m_pathsPruned = m_pathsPruned + 1;
33         continue;
34     else
35         pathTemp = [i pathStruct.path];
36         pathCost = pathStruct.cost + w(i);
37         c(i) = pathCost;
38
39         extendedPaths(end+1) = struct('path',{pathTemp},'cost',{pathCost});
40         m_extendedPathsCount = m_extendedPathsCount + 1;
41     end
42 end
```

```
function [optPath,runTime] ...
             = dynamicProgramming(x,y,r,n,aT,dt,SATELLITE)

%% ##########################
% dynamicProgramming.m
% ##########################
%
% This function finds the optimal path for imaging a series of ground
% targets using a dynamic programming algorithm.
%
%  INPUTS   x           Vector of target x-coordinates (length n). [km]
%           y           Vector of target y-coordinates (length n). [km]
%           r           Vector of target priority (length n). [ND]
%           n           Number of targets.
%           aT          Lookup table containing satellite agility
%                       information; each row is an [angle(deg), time(s)]
%                       data point, the time required for the
%                       satellite to slew the given angle.
%           dt          Time resolution. [s]
%           SATELLITE   Structure containing satellite and imaging area
%                       specifications. See 'Satellite Parameters' below.
%
%  OUTPUTS  optPath     Structure containing optimal path and cost.
%                           .path   Vector of targets to visit
%                           .cost   Optimal path value
%           runTime     Run time of the function call. [s]
%%

tic

global plotFlag
global dispFlag

%-------------------------------
% Optimization Parameters
%-------------------------------
testing = 0;                % code testing flag
debug = 0;                  % code debugging flag
rtd = 180/pi;               % conversion rad to deg
```

```
40 dtr = pi/180;                % conversion deg to rad
41 gamma = 1;                   % DP discount factor
42 epsilon = 0.01;              % DP convergence criterion
43 t_form = 'equally_spaced';   % set type of time discretization
44 reward_type = 'binary';      % set type of imaging reward
45 trans_time = 'dynamic';      % set transition time constant or dynamic
46
47 %-------------------------------
48 % Satellite Parameters
49 %-------------------------------
50 h = SATELLITE.h; %km            satellite altitute
51 eta = SATELLITE.eta; %deg       minimum elevation angle
52 v = SATELLITE.v; %km/s          orbital velocity
53 a = SATELLITE.a; %km            dimension of target area
54 b = SATELLITE.b; %km            dimension of target area
55 c = SATELLITE.c; %km            dummy start/end target offset
56 img_time = 0; %s                point imaging time
57
58 %-------------------------------
59 % Initialize Problem
60 %-------------------------------
61 x = x - x(1);    % normalize x coordinates to begin at zero
62 r(n) = 100;      % set reward for target N to be very large
63                  % so that optimal path ends at target N
64
65 % Satellite field of regard limits
66 rx = h/tan(eta*dtr);
67 rt = rx/v;
68 rt_ind = ceil(rt/dt);
69
70 % Time discretization
71 T = (a+2*c)/v;
72 if strcmp(t_form,'equally_spaced')
73     t = linspace(0,T,T/dt);
74 elseif strcmp(t_form,'zero_pitch')
75     t = x/v;
76 end
77
78 % Vector of satellite positions at chosen times
```

```
79  x_s = t*v;
80
81  % Required imaging time per point
82  t_img = img_time*ones(n,1);
83  t_img(1) = 0; t_img(end) = 0;
84
85  % Initialize storage structure for DP state information
86  STATES = struct('IMG_WINDOW',[]);
87
88  %-------------------------
89  % Find Feasible States
90  %-------------------------
91
92  for i = 2:n-1
93      % Calculate the time at which the satellite
94      % passes the target in x
95      t_ov(i) = (x(i)-x(1))/v;
96      % Find the index of t at which this occurs
97      t_ov_ind(i) = find(t - t_ov(i) >= 0,1,'first');
98      % Calculate the index bounds for which
99      % target i could possibly be in view
100     t_min_ind = max(1,t_ov_ind(i)-rt_ind);
101     t_max_ind = min(length(t),t_ov_ind(i)+rt_ind);
102
103     k_ind = []; k_rew = [];
104     % Iterate over all imaging opportunities for target i
105     for k = t_min_ind:1:t_max_ind
106         % If the satellite at t(k) is within range of target i
107         d = [x_s(k)-x(i) y(i)];
108         d_e = [-t_img(i)*v 0];
109         % Create an imaging state for this target-time pair
110         if ( norm(d) < rx && norm(d + d_e) < rx )
111             % Call reward function to calculate the value of this image
112             rew = reward(d,r(i),rx,reward_type);
113             % Add the time index and reward to the list for this target
114             k_ind = [k_ind k];
115             k_rew = [k_rew rew];
116         end
117     end
```

```
118     % Store all of the feasible imaging states for target i
119     STATES(i).IMG_WINDOW = k_ind;
120     STATES(i).IMG_REWARD = k_rew;
121     STATES(i).IMG_WINDOW_IND_SZ = length(k_ind);
122 end
123
124 % Designate the start and end imaging states
125 STATES(1).IMG_WINDOW = 1;
126 STATES(1).IMG_REWARD = r(1);
127 STATES(1).IMG_WINDOW_IND_SZ = 1;
128 STATES(n).IMG_WINDOW = length(t);
129 STATES(n).IMG_REWARD = r(n);
130 STATES(n).IMG_WINDOW_IND_SZ = 1;
131
132 % Record intermediate runtime
133 toc1 = toc;
134
135 if testing == 1
136     % Display imaging states for each target
137     figure
138     % Display target set in x-y plane
139     subplot(2,1,1)
140     scatter(x,y,'s','b');
141     xlabel('along-track direction (km)')
142     ylabel('cross-track direction (km)')
143     subplot(2,1,2)
144     xlim([0 t(end)])
145     xlabel('time (s)')
146     ylabel('target')
147     hold on
148     colormap(cool);
149     for i = 1:n
150         % For each target, display the set of imaging times
151         vis = t(STATES(i).IMG_WINDOW);
152         scatter(vis,-i*ones(length(vis),1),'s')
153     end
154 end
155
156 %-------------------------------------
```

```
% Find Feasible State Transitions
%-----------------------------------------
if strcmp(trans_time,'dynamic')
    for i = 1:n
        % Get possible imaging times for target i
        t_i = STATES(i).IMG_WINDOW;
        % Loop over possible imaging times
        for k = 1:length(t_i)
            t_i_k = t_i(k);
            % Calculate satellite - target vector
            % for target i at time t(k)
            r_i_k = [x(i)-x_s(t_i_k) y(i) -h];
            % Modify to account for imaging time
            r_i_k = r_i_k + [-t_img(i)*v 0 0];
            % Loop over all targets ahead of target i
            for j = i+1:n
                % For target j, see if there is an
                % imaging opportunity after time t(k)
                t_j_loc = find(STATES(j).IMG_WINDOW > t_i_k);
                % If there is, determine feasibility
                if(~isempty(t_j_loc))
                    t_j = STATES(j).IMG_WINDOW(t_j_loc);
                    % Loop over all possible imaging times for target j
                    m_feas = [];
                    for m = 1:length(t_j)
                        t_j_m = t_j(m);
                        % Calculate satellite - target
                        % vector for target j at time t(m)
                        r_j_m = [x(j)-x_s(t_j_m) y(j) -h];
                        % Calculate repositioning angle required
                        phi = acos(dot(r_i_k,r_j_m) ...
                                /(norm(r_i_k)*norm(r_j_m)));
                        % Lookup time required
                        s = agilityLookup(rtd*phi,aT);
                        % If sufficient time to reposition
                        if s + t(t_i_k) + t_img(i) ≤ t(t_j_m)
                            % State transition action is legal
                            m_feas = [m_feas t_j_loc(m)];
                        end
```

```
196                     end
197                     % Update state transitions for i-k to j
198                     STATES(i).ACTIONS(k).target(j).trans = m_feas;
199                 end
200             end
201         end
202     end
203 end
204
205 %-----------------------------------------
206 % Find Feasible State Transitions
207 %   CONSTANT TRANSITION TIME ASSUMPTION
208 %-----------------------------------------
209
210 if strcmp(trans_time,'constant')
211     % Calculate approximate constant transition times
212     for i = 1:n
213         for j = i+1:n
214             r_i = [0 y(i) -h];
215             r_j = [0 y(j) -h];
216             % Calculate repositioning angle required
217             phi = acos(dot(r_i,r_j)/(norm(r_i)*norm(r_j)));
218             % Lookup time required
219             s(i,j) = agilityLookup(rtd*phi,aT);
220         end
221     end
222
223     for i = 1:n
224         % Get possible imaging times for target i
225         t_i = STATES(i).IMG_WINDOW;
226         % Loop over possible imaging times
227         for k = 1:length(t_i)
228             t_i_k = t_i(k);
229             % Loop over all targets again
230             for j = i+1:n
231                 % For target j, see if there is an
232                 % imaging opportunity after time t(k)
233                 t_j_loc = find(STATES(j).IMG_WINDOW > t_i_k);
234                 % If there is, determine feasibility
```

```
235                 if(¬isempty(t_j_loc))
236                     t_j = STATES(j).IMG_WINDOW(t_j_loc);
237                     % Loop over all possible imaging times for target j
238                     m = 0;
239                     bool_c = 1; % Store whether a legal transition is found
240                     while bool_c && m < length(t_j)
241                         m = m + 1;
242                         t_j_m = t_j(m);
243                         % If sufficient time to reposition
244                         if t(t_i_k) + t_img(i) + s(i,j) ≤ t(t_j_m)
245                             % State transition action is legal
246                             bool_c = 0;
247                         end
248                     end
249                     % Once the first legal transition is found, the
250                     % remaining transitions to t_j_m are also feasible
251                     if bool_c == 0 && m ≤ length(t_j)
252                         m_feas = t_j_loc(m:end);
253                     else
254                         m_feas = [];
255                     end
256                     % Update state transitions for i-k to j
257                     STATES(i).ACTIONS(k).target(j).trans = m_feas;
258                 end
259             end
260         end
261     end
262 end
263
264 % Record intermediate runtime
265 toc2 = toc - toc1;
266
267 %-------------------------------------------------------------
268 % Arrange Dynamic Programming Variables
269 %-------------------------------------------------------------
270 % Create a single array of states for DP implementation
271
272 % Find the total number of states and create an index to store where
273 % each state begins in the DP array
```

```
target_index = zeros(n,1);
target_index(1) = 1;
for i = 2:n
    target_index(i) = target_index(i-1) + STATES(i-1).IMG_WINDOW_IND_SZ;
end
n_states = target_index(n) + STATES(n).IMG_WINDOW_IND_SZ - 1;

% Create reward array (for state transitions)
rew = [];
for i = 1:n
    rew = [rew; STATES(i).IMG_REWARD'];
end

% Create appropriate state references
% Loop over all targets
for i = 1:n
    % Loop over all possible i-k states for that target
    for k = 1:length(STATES(i).ACTIONS)
        % Initialize an array to store references
        state_pointers = [];
        % Iterate over all targets again
        % for j = 1:n
        for j = i+1:n
            if i ≠ j
                % Calculate where legal state transitions are in terms of
                % an array of states indexed by target via target_index.
                if(¬isempty(STATES(i).ACTIONS(k).target(j)))
                    if debug
                        disp(['i = ' num2str(i) ...
                            '; j = ' num2str(j) '; k = ' num2str(k)])
                    end
                    % Retrieve timing information about target j
                    t_k_loc = STATES(i).ACTIONS(k).target(j).trans;
                    % Store the reference information in state_pointers
                    state_pointers = [state_pointers ...
                                        (target_index(j)-1 + t_k_loc)];
                end
            end
        end
```

```
313          % Update the state transitions with reference information
314          STATES(i).ACTIONS(k).STATE_POINTERS = state_pointers;
315      end
316  end
317
318  % Record intermediate runtime
319  toc3 = toc - toc2;
320
321  %-------------------------------------------
322  % Dynamic Programming
323  %-------------------------------------------
324
325  % Initialize relavant parameters
326  % V and V_upd store the value of each state
327  % argMax stores the best action which can be taken
328  % for each state.
329  V = ones(n_states,1);
330  V_upd = zeros(n_states,1);
331  argMax = zeros(n_states,1);
332
333  if dispFlag
334      disp(['Number of DP States:' num2str(n_states)])
335  end
336
337  % While the algorithm has not converged, do
338  while norm(abs(V - V_upd)) > epsilon
339      % Reset V from V_upd
340      V = V_upd;
341      V_upd = zeros(n_states,1);
342      % Calculate value update before iterating over each state
343      V_comp = rew + gamma*V;
344      % For each target and each action
345      for i = 1:n-1
346          if ~isempty(STATES(i).ACTIONS)
347              % for k = 1:STATES(i).IMG_WINDOW_IND_SZ
348              for k = 1:length(STATES(i).ACTIONS)
349                  % Store the index of this state into the V vector
350                  state_index = target_index(i) + k - 1;
351                  % Retrieve possible state transitions
```

```
352                 transitions = STATES(i).ACTIONS(k).STATE_POINTERS;
353                 if debug
354                     disp(['i = ' num2str(i) '; k = ' ...
355                         num2str(k) '; STATE INDEX:' num2str(state_index)])
356                     disp(transitions')
357                 end
358                 % If any transitions are possible
359                 if ~isempty(transitions)
360                     % Get the array of action value, will be maximized over
361                     act = V_comp(transitions);
362                     % Find the maximum and where it occurs
363                     [V_upd(state_index),loc] = max(act);
364                     % Update the argMax with the appropriate action indexed
365                     % into V rather than act
366                     argMax(state_index) = transitions(loc);
367                 end
368             end
369         end
370     end
371 end
372
373 V = V_upd;
374 clear V_upd
375
376 % Record intermediate runtime
377 toc4 = toc - toc3;
378
379 %-------------------------------------
380 % Extract Optimal Path
381 %-------------------------------------
382
383 % optStates will hold the ordering of states to
384 % visit from state 1 (target 1). The variable k
385 % is used to navigate through argMax.
386 k = 1;
387 optStates(1) = 1;
388 % While we haven't reached the goal state
389 while k < n_states
390     % Find the next consecutive state using argMax
```

```
391      optStates = [optStates argMax(k)];
392      % Store the following state
393      k = argMax(k);
394  end
395
396  % Convert array of states to i-k pairs
397  optPath = zeros(length(optStates),1);
398  optImgTime = zeros(length(optStates),1);
399  for i = 1:length(optStates)
400
401      % Find which target j corresponds to this state
402      if optStates(i) < n_states
403          j = find(target_index > optStates(i),1,'first')-1;
404      else
405          j = n;
406      end
407      % Find which IMG_WINDOW for j is being used
408      k = optStates(i) - target_index(j) + 1;
409      % Retrieve the time index for this IMG_WINDOW
410      t_ind = STATES(j).IMG_WINDOW(k);
411      % Store optimal path information
412      optPath(i) = j;
413      optImgTime(i) = t_ind;
414  end
415
416  % Subtract the artificial reward for target N
417  cost = V(1)-r(end);
418
419  % Record total runtime
420  runTime = toc;
421  toc5 = runTime - toc4;
422
423  %-----------------------------------
424  % Runtime Analysis
425  %-----------------------------------
426  if 0
427      figure
428      colormap('cool');
429      barh([toc1;toc2;toc3;toc4;toc5],'stack')
```

```
430 end
431
432 %-------------------------------
433 % Plot Results
434 %-------------------------------
435 if plotFlag
436     plotPath(x,y,optPath,cost,t(optImgTime),SATELLITE);
437 end
438
439 % Create struct of optimal path to return
440 optPath = struct('path',{optPath'},'cost',cost);
441 end
442
443 %-------------------------------
444 % Reward Function
445 %-------------------------------
446 function r = reward(d,r_i,rx,reward_type)
447     % INPUT     d       2D vector to target
448     %           r_i     Target i priority
449     %           rx      Ground range
450     % OUTPUT    r       Calculated reward
451     if strcmp(reward_type,'binary')
452         r = r_i;     % Reward = target priority
453     elseif strcmp(reward_type,'linear')
454         % Reward = scaled based on range to target
455         r = r_i*(1 - 0.1*abs(d(1))/rx);
456     elseif strcmp(reward_type,'zero_pitch')
457         % Reward only images taken without
458         % satellite pitching motion
459         if abs(d(1)) ≤ 0.001
460             r = r_i;
461         else
462             r = 0;
463         end
464     end
465 end
```

```
function plotPath(x,y,path,cost,imgTime,SATELLITE)

%% %%%%%%%%%%%%%%%%%%%%%%
% plotPath.m
% %%%%%%%%%%%%%%%%%%%%%%
%
% This function plot the optimal path for imaging a series of ground
% targets.
%
%  INPUTS   x            Vector of target x-coordinates (length n). [km]
%           y            Vector of target y-coordinates (length n). [km]
%           path         Vector of optimal path.
%           cost         Scalar optimal cost.
%           imgTime      Time at which each image is taken, referenced to
%                        t = 0 at target 1.
%           SATELLITE    Structure containing satellite and imaging area
%                        specifications. See 'Satellite Parameters' below.
%%

%-----------------------------------
% Satellite Parameters
%-----------------------------------
a = SATELLITE.a; %km            dimension of target area
b = SATELLITE.b; %km            dimension of target area
c = SATELLITE.c; %km            dummy start/end target offset
v = SATELLITE.v; %km/s          orbital velocity

% Get coordinates of optimal path targets
xOpt = x(path);
yOpt = y(path);
x_s = imgTime*v;

figure
hold on

% Color palette
C2 = [0.0476    0.9524    1.0000];
C3 = [     0         0    1.0000];

```

```
% Plot target set and imaged targets
scatter(x,y,[],C2,'s')
scatter(xOpt,yOpt,'m','s','filled')
scatter(x_s,zeros(length(xOpt),1),[],C3)

% Plot satellite-target connecting line for each image
for i = 1:length(xOpt)
    hold on
    plot([xOpt(i) x_s(i)],[yOpt(i) 0],'k-')
end

% Set plot dimensions
ylim(4/3*b*[-1 1])
xlim([0  a+2*c])
% Plot imaging area boundaries
plot(linspace(0,a+2*c,5), b*ones(5,1),'k--')
plot(linspace(0,a+2*c,5),-b*ones(5,1),'k--')

xlabel('along-track direction (km)')
ylabel('cross-track direction (km)')
title(['Solution Path, Cost = ' num2str(cost)])
```

Bibliography

[1] Lighter-than-air systems for future naval missions. Naval Research Advisory Committee. The Pentagon Auditorium, Washington, D.C., October 2005.

[2] Matlab version 7.1. The MathWorks, Inc., August 2005.

[3] Satellite tool kit version 7.1. Analytical Graphics Incorporated, June 2006.

[4] SC Modeler version 2.9. AVM Dynamics, February 2008.

[5] W. S. Adams and L. Rider. Circular polar constellations providing continuous single or multiple coverage above a specified latitude. *Journal of the Astronautical Sciences*, 35:155–192, June 1987.

[6] David A. Bearden. Small satellite costs. The Aerospace Corporation, El Segundo, California, `http://www.aero.org/publications/crosslink/winter2001/04.html`, October 2004.

[7] E. Bensana, G. Verfaillie, J. Agnse, N. Bataille, and D. Blumstein. Exact and inexact methods for the daily management of an earth observation satellite. In *Proc. 4 th Intl. Symposium on Space Mission Operations and Ground Data Systems*, Munich, Germany, 1996.

[8] Dimitri P. Bertsekas, editor. *Dynamic Programming and Optimal Control, Vol. I.* Optimization and Computation. Athena Scientific, Belmont, Massachusetts, third edition, 2005.

[9] Dimitris Bertsimas and John N. Tsitsiklis, editors. *Introduction to Linear Optimization.* Optimization and Computation. Athena Scientific, Belmont, Massachusetts, first edition, 1997.

[10] Carl Blaurock. Disturbance-optics-controls-structures (docs). Technical report, Night-sky Systems, Inc., 2006.

[11] John M. Borky and Robert E. Conger. Operational concepts and payoffs for responsive space systems. In *Proc. 1st Responsive Space Conference*, Redondo Beach, CA, April 2003. AIAA.

[12] Marco Cáceres. Cost overruns plague military satellite programs. Teal Group, Aerospace America, January 2006.

[13] Reynold W. Cochran and Richard H. Vassar. Fast-steering mirrors in optical control systems. *Advances in Optical Structure Systems*, 1303(1):245–251, 1990.

[14] Lucy E. Cohan. Integrated modeling to facilitate control architecture design for lightweight space telescopes. Ssl report #11-07, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, `http://ssl.mit.edu/publications/theses/SM-2007-CohanLucy.pdf`, June 2007.

[15] Lucy E. Cohan, Elizabeth O. Jordan, and David W. Miller. Tradespace exploration for a parameterized modular optical space telescope (most). In Wizard V. Oz and Mihalis Yannakakis, editors, *Proc. 48th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, All AIAA Conferences, Honolulu, Hawaii, April 2007.

[16] Antonio J. Conejo, E. Castillo, R. Mínguez, and R. García-Bertrand, editors. *Decomposition Techniques in Mathematical Programming: Engineering and Science Applications.* Springer, Berlin, New York, first edition, 2006.

[17] John E. Draim, R. Inciardi, Paul Cefola, R. Proulx, and D. Carter. Demonstration of the cobra teardrop concept using two smallsats in 8-hour elliptical orbits. In *Proc. 15th Annual USU Conference on Small Satellites*, Logan, UT, August 2001.

[18] Thomas A. Young et al. Report of the defense science board/air force scientific advisory board joint task force on acquisition of national security space programs. Teal Group, Aerospace America, Washington, D.C., `http://www.fas.org/spp/military/dsb.pdf`, May 2003.

[19] Virginie Gabrel. Strengthened 0-1 linear formulation for the daily satellite mission planning. *Journal of Combinatorial Optimization*, 11(3):341–346, May 2006.

[20] Virginie Gabrel, Alain Moulet, Cécile Murat, and Vangelis Th. Paschos. A new single model and derived algorithms for the satellite shot planning problem using graph theory concepts. *Annals of Operations Research*, 69:115–134(20), 1997.

[21] Virginie Gabrel and Cécile Murat. *Mathematical Programming for Earth Observation Satellite Mission Planning.* Applied Optimization. Kluwer Academic Publishers, The Netherlands, first edition, 2003.

[22] Virginie Gabrel and Daniel Vanderpooten. Enumeration and interactive selection of efficient paths in a multiple criteria graph for scheduling an earth observing satellite. *European Journal of Operational Research*, 139(3):533–542, June 2002.

[23] Global hawk factsheet. U.S. Air Force, Aeronautical Systems Center, `http://www.af.mil/factsheets/factsheet.asp?fsID=175`, October 2005.

[24] Al Globus, James Crawford, Jason Lohn, and Anna Pryor. A comparison of techniques for scheduling earth observing satellites. In *Proc. Sixteenth Innovative Applications of Artificial Intelligence Conference (IAAA)*, number 19 in All IAAA Conferences, pages 836–843. IAAA, Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2004.

[25] Google Maps. `http://maps.google.com`, May 2008.

[26] Djamal Habet and Michel Vasquez. Saturated and consistent neighborhood for selecting and scheduling photographs of agile earth observing satellite. In *MIC2003: Proc. of the 5th Metaheuristics International Conference*, pages 28–1–28–6, Kyoto, Japan, August 2003.

[27] Djamal Habet and Michel Vasquez. Solving the selecting and scheduling satellite photographs problem with a consistent neighborhood heuristic. In *ICTAI '04: Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence*, pages 302–309, Washington, DC, USA, 2004. IEEE Computer Society.

[28] Nicholas G. Hall and Michael J. Magazine. Maximizing the value of a space mission. *European Journal of Operational Research*, 78(2):224–241, October 1994.

[29] S. Harrison and M. Price. Task scheduling for satellite based imagery. In *Dans Proceedings of the Eighteenth Workshop of the UK Planning and Scheduling Special Interest Group, pages 64-78, University of Salford, UK, December 1999.*

[30] IKONOS. European Space Imaging, Inc. `http://www.ikonos.be.`, April 2008.

[31] GeoEye Imagery Products: IKONOS. `http://www.geoeye.com/products/imagery/ikonos/default.htm`, May 2008.

[32] NASA Goddard Space Flight Center. News Release, August 24, 2007. `http://www.nasa.gov/centers/goddard/news/topstory/2007/wavefront.html`, April 2008.

[33] Cyrus D. Jilla. *A Multiobjective, Multidisciplinary Design Optimization Methodology for the Conceptual Design of Distributed Satellite Systems.* Ssl report #6-02, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, `http://ssl.mit.edu/publications/theses/PhD-2002-JillaCyrus.pdf`, May 2002.

[34] Thomas A. Johnson, Jeffery Antol, Ted M. Holtz, and Vince Cuda. Using fast-steering mirror control to reduce instrument pointing errors caused by spacecraft jitter. NASA Contractor Report 201583, June 1996.

[35] Brian L. Kantsiper, Patrick A. Stadter, and Pamela L. Stewart. Ors heo constellations for continuous availability. In *Proc. 5th Responsive Space Conference*, Los Angeles, CA, April 2007. AIAA.

[36] The Landsat Program. NASA. `http://landsat.gsfc.nasa.gov/`, May 2008.

[37] Scott C. Larrimore. Partially continuous earth coverage from a responsive space constellation. In *Proc. 5th Responsive Space Conference*, Los Angeles, CA, April 2007. AIAA.

[38] Wiley J. Larson and James R. Wertz, editors. *Space Mission Analysis and Design.* Space Technology. Microcosm Press, El Segundo, CA, third edition, 1999.

[39] Michel Lemaître, Gérard Verfaillie, Frank Jouhaud, Jean-Michel Lachiver, and Nicolas Bataille. Selecting and scheduling observations for agile satellites. *Aerospace Science and Technology*, 6(5):367–381, September 2002.

[40] Da-Yin Liao and Yu-Tsung Yang. Imaging order scheduling of an earth observation satellite. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(5):794–802, September 2007.

[41] Wei-Chen Lin and Da-Yin Liao. A tabu search algorithm for satellite image scheduling. In *Proc. of the IEEE International Conference on Systems, Man, and Cybernetics*, number 2 in All IEEE Conferences, pages 1601–1606, The Hague, Netherlands, October 2004. IEEE.

[42] Wei-Cheng Lin, Da-Yin Liao, Chung-Yang Liu, and Yong-Yao Lee. Daily imaging scheduling of an earth observation satellite. *IEEE Transactions on Systems, Man, and Cybernetics*, 35(2):213–223, March 2005.

[43] C. Mancel and P. Lopez. Complex optimization problems in space systems., February 2008.

[44] William Martin. Satellite image collection optimization. *Optical Engineering*, 41(9):2083–2087, September 2002.

[45] Col. Kevin McLaughlin. Operationally Responsive Space Office. `http://www.responsivespace.com/ors/reference/McLaughlin.pdf`, July 2007.

[46] A. B. Meinel. Cost Scaling Laws Applicable to Very Large Optical Telescopes. In D. L. Crawford, editor, *Instrumentation in Astronomy III*, volume 172 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 2–+, 1979.

[47] Space Systems Lab. Massachusetts Institute of Technology. `http://ssl.mit.edu/`, May 2008.

[48] New dod programs can learn from past efforts to craft better and less risky acquisition strategies. United States Government Accountability Office GAO-06-447, March 2006.

[49] RQ-4 Block 10 Global Hawk. Northrop Grumman Integrated Systems. `http://www.is.northropgrumman.com/systems/ghrq4a.html`, May 2008.

[50] Simon Nolet, Edmund Kong, and David W. Miller. Autonomous docking algorithm development and experimentation using the spheres testbed. volume 5419, pages 1–15. SPIE, 2004.

[51] Joseph C. Pemberton and III Flavius Galiber. A constraint-based approach to satellite scheduling. In *DIMACS workshop on on Constraint programming and large scale discrete optimization*, pages 101–114, Boston, MA, USA, 2001. American Mathematical Society.

[52] Responsive Space Conference. `http://www.responsivespace.com/`, May 2008.

[53] Joseph Rouge. Operationally responsive space: Implementing the way ahead. In *Proc. 5th Responsive Space Conference*, Los Angeles, CA, April 2007. AIAA.

[54] Stuart Russell and Peter Norvig, editors. *Artificial Intelligence: A Modern Approach.* Artificial Intelligence. Prentice Hall, Upper Saddle River, New Jersey, second edition, 2003.

[55] Niek Schaap. Ikonos: future and present. volume 4881, pages 660–668. SPIE, 2003.

[56] Uriel Scialom. Optimization of satellite constellation reconfiguration. Ssl report #14-03, Massachusetts Institute of Technology, Dept. of Aeronautics and Astronautics, `http://ssl.mit.edu/publications/theses/SM-2003-ScialomUriel.pdf`, August 2003.

[57] SPOT Image Corporation. `http://www.spot.com/`, May 2008.

[58] SPOT. Spot image technical data. `http://www.spot.com/web/SICORP/445-sicorp-the-spot-satellites.php`, March 2008.

[59] Jeff Summers, Greg Heinsohn, and Greg Hegemann. 'launch-on-demand'; a revolutionary paradigm for space utilization. In *Proc. 1st Responsive Space Conference*, Redondo Beach, CA, April 2003. AIAA.

[60] S.A. Uebelhart, D. Howell, and D.W. Miller. Evaluating alternative architectures for lightweight space telescopes using parameterized models. *Aerospace Conference, 2006 IEEE*, pages 12 pp.–, 4-11 March 2006.

[61] Scott A. Uebelhart, Lucy E. Cohan, and David W. Miller. Design exploration for a modular optical space telescope architecture using parameterized integrated models. In *Proc. 47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*. AIAA, 2006.

[62] Michel Vasquez and Jin-Kao Hao. A 'logic-constrained' knapsack formulation and a tabu algorithm for the daily photograph scheduling of an earth observation satellite. *Computational Optimization and Applications*, 20:137–157(21), November 2001.

[63] Michel Vasquez and Jin-Kao Hao. Upper bounds for the spot 5 daily photograph scheduling problem. *Journal of Combinatorial Optimization*, 7(1):87–103, March 2003.

[64] J.G. Walker. Continuous whole-earth coverage by circular-orbit satellite patterns. Royal Aircraft Establishment Technical Report 77044.

[65] James R. Wertz, editor. *Mission Geometry: Orbit and Constellation Design and Management*. Space Technology. Microcosm Press, El Segundo, CA, first edition, 2002.

[66] James R. Wertz. Coverage, responsiveness, and accessibility for various 'responsive orbits'. In *Proc. 3rd Responsive Space Conference*, Los Angeles, CA, April 2005. AIAA.

[67] James R. Wertz, Richard E. Van Allen, and Christopher J. Shelner. Aggressive surveillance as a key application for responsive space. In *Proc. 4rd Responsive Space Conference*, Los Angeles, CA, April 2006. AIAA.

[68] William J. Wolfe and Stephen E. Sorensen. Three scheduling algorithms applied to the earth observing systems domain. *Management Science*, 46(1):148–166, January 2000.

[69] Content Collection Systems. DigitalGlobe Inc. `http://www.digitalglobe.com/index.php/86/WorldView-1`, April 2008.